



Universidad
Carlos III de Madrid

Departamento de Teoría de la Señal y Comunicaciones

PROYECTO FIN DE CARRERA

ANÁLISIS, DISEÑO Y DESARROLLO DE UN CRM PARA UNA PYME

Autor: Daniel Torbellino Rubio

Tutor: Matilde Sánchez Fernández

Leganés, 28 de octubre de 2013

Título: Análisis, Diseño y Desarrollo de un CRM para una PYME

Autor: Daniel Torbellino Rubio

Tutor: Matilde Sánchez Fernández

EL TRIBUNAL

Presidente: Harold Molina Bulla

Vocal: M^a del Carmen Fernández Panadero

Secretario: Raquel Pérez Leal

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 28 de Octubre de 2013 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de:

VOCAL

SECRETARIO

PRESIDENTE

"Saber demasiado nunca simplifica las decisiones."

Dune

"El requisito del éxito es la prontitud en las decisiones."

Sir Francis Bacon

Agradecimientos

Lo primerísimo, a las mujeres importantes de mi vida, a mi novia, Noelia, que no ha parado de apoyarme para que acabe mi carrera luchando con mi activa pasividad; y sobre todo a mi madre, que la debo absolutamente todo lo que he sido, soy y seré en la vida y a la que no voy a poder besar jamás siendo ingeniero, algo que no podré perdonarme.

A mi familia, por supuesto, que cada reunión familiar me recordaban mi deber para con Atenea...en especial a mi hermana de quien no he sabido ser mejor hermano y a mi padre, de cuyo sudor comenzó a florecer mi carrera.

A mis amigos, Sergio y David, que aquella noche en San Martín supieron darme el último empujón entre zarzas, insectos, animales y quien sabe que más ánimas nos encontramos entre castaños.

Y por último, a todos los profesores que me han ido llevando de mano en mano y convirtiendo mi curiosidad endémica en una ingeniería aplicada...a saber y sabiendo que no puedo recoger a todos, Lola, M^a Teresa, “El pipa” y por supuesto a mi tutora, Mati.

Resumen

El proyecto, presenta el análisis, diseño y desarrollo de un CRM (Customer Relationship Management) adaptado para una PYME (Pequeña Y Mediana Empresa) del sector del alquiler de maquinaria cuyo nombre comercial sería rmN, donde realicé casi el 100% del trabajo.

Un CRM es la herramienta comercial y de marketing clave en una compañía. Como repasaremos en el estado del arte, es el pilar donde gira la acción de captación de cliente de una empresa y donde analizan el efecto de sus acciones de marketing. En aquel momento (2002) no existía una solución de bajo coste por lo que este producto tenía cuota en el mercado. Veremos por diferentes analistas como está premisa se ha hecho realidad en una década siendo desbancados los SAP, Oracle o Microsoft por soluciones tipo salesforce.com.

Sobre la base tecnológica de JAVA, el proyecto se planteó con la idea de que este desarrollo a medida para la empresa MERA, pudiera constituirse como un producto dentro de la compañía Planet Media para poder ser comercializado para otros clientes y sectores, por lo que uno de los criterios de diseño fue la modularidad del sistema así como la facilidad de adecuación y una arquitectura sencilla pero robusta.

Además de los criterios técnicos que nos llevaron al uso de Java como tecnología, se llevó a cabo una toma de requisitos con el cliente que debía cumplir la herramienta como fueron gestión de roles de acceso, manejo y clasificación de clientes, envío de newsletter mensuales, estadísticas sobre clientes, gestión de campañas y alertas inteligentes.

Cada uno de estos requisitos constituyó un reto dentro la de la programación que se detalla en esta memoria.

Adicionalmente, para el módulo de estadísticas, decidimos aplicar conceptos vistos en Tratamiento Digital de la Información para no dar sólo el dato, si no información que pudiera ayudar a decidir donde poner foco en las campañas a cliente y sobre el cual se centra uno de los capítulos de esta memoria.

Palabras clave:

Java, CRM, TDI, Tratamiento Digital de la Información, Decisión

Abstract

The project, as its title suggests, presents the analysis, design and development of a CRM (Customer Relationship Management) adapted for a SMEs (Small and Medium Enterprise) sector of machinery hire which commercial name was rmN, where I did almost 100 % of the work.

A CRM is sales and marketing key tool for every company. As we will review on the state of the art, a CRM is the system where resides all the new business development knowledge of a company and where they analyse the effect of their marketing actions. “Once upon a time” (the project finished on 2002) there was no low cost solution to this kind of product within the market share. We will present different research results about CRM evolution concluding that in just a decade, enterprise solutions such as SAP and Oracle have been supplanted by SMEs solution, with an on demand devil very model such as salesforce.com.

Built on Java technology, the project was designed with the idea to create a Product, based on this development for the company MERA, and could establish itself as a product within Planet Media offering. The modularity of the system and the ease of adjustment and simple but robust architecture was a key premise due to Planet Media strategy relies on a multi sector CRM solution.

In addition to the technical criteria that led us to the use of Java as a technology, it was collected a number of requirements with the customer such as: role access management, customer classification and management, newsletter in a monthly basis, statistics about customers, campaign management and intelligent alerts.

Each of these requirements was a challenge in the programming part of the project, explained much more detailed ahead.

Additionally, for the analytics module, we decided to apply concepts covered within Digital Information Processing, helping to make marketing investment decision to the customer.

Keywords:

Java, CRM, Digital Information Processing, Decition Making

Advertencia

ADVERTENCIA: En este texto se utilizan con profusión términos directamente tomados del inglés, sobre todo para nombrar dispositivos, denominaciones técnicas o conceptos propios de las telecomunicaciones.

Se conocen las reglas ortográficas del español respecto a citar entre comillas o en cursiva los términos comunes de otras lenguas.

A nuestro entender, en trabajos como este, el uso más que frecuente de esos signos de distinción dificultaría la lectura, destinada además a lectores que conocen de sobra estos términos y no precisan de esa diferenciación

Prefacio

Hace ya más de 10 años que terminé con todas las asignaturas de mi carrera y finalicé mi proyecto fin de carrera, por lo que ha pasado mucho tiempo desde que pusimos en producción este CRM y el momento en el que estoy escribiendo esta memoria, por tanto, habrá muchas de las cosas que aquí recojo, sobre todo técnicamente, que fueron una innovación, un reto hace una década y que ahora no son más que obsolescencias, por ello quiero pedir disculpas a la audiencia de esta memoria antes de continuar con ella.

Lo que si creo que sigue vigente y de lo que puedo seguir sintiéndome orgulloso es la motivación que me llevo a realizar este proyecto fin de carrera: **aplicar la ingeniería al mundo real**.

Con esta motivación, mi proyecto no podría discurrir en ningún sitio mejor que en una empresa, y por ello moví tanto a mi directora de proyecto como al dueño de la que era mi empresa para que juntos pudiéramos dar con algo que sirviera mi objetivo, la aplicación de lo aprendido para ayudar a una empresa.

Dicho y hecho, una vez hablado con ambas partes, llegamos rápido al consenso, desarrollaría de manera íntegra un nuevo proyecto que mi empresa acaba de ganar, un CRM a medida para una PYME dedicada al alquiler de maquinaria de construcción.

Lógicamente, siendo programación orientada a objetos, podía aplicar lo aprendido durante los primeros años de la carrera, pero no me valía sólo con aplicar lo aprendido, quería evolucionar, completar mi educación no sólo con experiencia si no con nuevos retos. Por aquel entonces, estaba cogiendo fuerza la orientación web frente a la cliente-servidor por lo que tuve que adquirir conocimientos de HTML, JSP, Servlet no visto en la carrera. Adicionalmente, la separación en capas era un concepto básico para poder reutilizar e independizar el negocio de la técnica, y con especial atención a las BBDD donde tuve que aprender SQL y el uso del API JDBC. Todos estos conceptos serán explicados debidamente en el estado del arte.

Con esto me sumergía en dos retos. Por un lado el técnico, la programación en sí misma y la aplicación a un contexto real de los conceptos de decisores inteligentes. Por otro el contexto, la realidad a modelar a través de la técnica, en este caso procesos de

marketing, de la relación con cliente, algo con lo que no estaba muy familiarizado y que ahora es parte de mi día a día.

Siendo un proyecto real, me sirvió también para ampliar mis capacidades de Gestión de Proyectos y siendo mi empresa de menos de 5 empleados, también adquirí mucho conocimiento sobre Gestión y Dirección de empresas y algo que se ha ido convirtiendo en mi vocación, la Gestión de Clientes y de Personas.

Pero me faltaba algo, en mi carrera esos tres conceptos no son troncales, necesitaba una guinda que cerrara mi ciclo universitario, y esta fue una de las asignaturas que más me ha marcado y que ha día de hoy me sigue fascinando, sigo aplicando y no puedo dejar de ver como se pueden mejorar múltiples procesos si la utilizáramos más.

Este área de conocimiento es el Tratamiento Digital de la Información, que durante mi etapa de estudiante me impartió el Director de la Titulación, el Catedrático Aníbal R. Figueiras-Vidal.

En el contexto del Marketing, la aplicación que se me ocurrió fue ayudar a esta empresa a decidir de forma inteligente, basándonos en los datos y la información de los clientes, sobre cuales se debía realizar una actuación puntual y sobre cuales invertir más en ellos.

Utilizaríamos la información recogida dentro del CRM por el uso de la extranet, las campañas y las newsletter para decidir sobre cuales clientes convenía poner foco.

Sin duda se podía haber incluido mucha más información, más parámetros, utilizar mecanismos más complejos que la regresión lineal, pero para una primera aproximación, nos pareció que daba suficiente beneficio y que me permitía aplicar suficiente los conceptos adquiridos durante la carrera.

Cual ha sido mi sorpresa que documentándome para la memoria (dejé el mundo de la programación muy poco después del proyecto) que eso que aplicamos en su momento se ha ido convirtiendo en una de las claves de CRM...todo un impacto!

Creo que con esto perfilo bastante lo que va a ser el contenido del proyecto que aquí recojo y cuales fueron las motivaciones que me llevaron a realizarlo, de hecho, es posible que me haya extendido de más y que mi lenguaje no haya sido todo lo científico que merece un tema como el tratado para una entidad como es la Universidad, trataré de mejorar en adelante.

Índice general

1. INTRODUCCIÓN Y OBJETIVOS.....	1
1.1 Contexto	1
1.2 Objetivos y Motivación.....	3
1.3 Estructura de la memoria.....	5
2. ESTADO DEL ARTE	6
2.1 CRM	6
2.1.1 Concepto de CRM.....	6
2.1.2 Ciclo de vida de la solución CRM	8
2.1.3 Clasificación de Clientes Dentro de las Soluciones CRM.....	10
2.1.3.1 Valor de la vida útil del cliente.....	10
2.1.3.2 Espectro de Mercado del cliente	11
2.1.3.3 Otras Métricas De Uso Habitual:	12
2.1.3.4 Minería de Datos	13
2.1.4 Espacio de Mercado de soluciones CRM	14
2.2 Tecnología Utilizada	18
2.2.1 Qué es Java.....	19
2.2.2 ¿Por qué JAVA?.....	20
2.2.3 Bases de datos distribuidas (JDBC)	25
2.2.3.1 Descripción del API JDBC.....	25
2.2.3.2 Arquitectura de JDBC	25
2.2.3.3 Apoyo del Mercado.....	27
2.2.3.4 Ventajas de la tecnología de JDBC.....	27
2.2.3.5 Características Clave	28
2.2.3.6 Conectividad de Bases de Datos de Java (JDBC).....	28
2.2.4 Java e Internet: jsp y servlet.....	29
2.2.4.1 ¿Qué es JSP?	30
2.2.4.2 ¿Cuáles son las Ventajas de JSP?	30
2.2.4.3 ¿Qué son los JSP?	31
2.3 Tratamiento Digital De La Información.....	32
2.4 Conclusiones Del Estado Del Arte	35
3. DISEÑO Y DESARROLLO DEL SISTEMA	36
3.1 Introducción	36
3.2 Diseño Funcional del Sistema	37
3.2.1 Caso de estudio.....	37
3.2.2 Modelo Funcional.....	38
3.2.3 rmN Administrador.....	39
3.2.4 rmN Contactos	40
3.2.5 rmN Campañas	41

3.2.6 <i>rmN Newsletter</i>	42
3.2.7 <i>rmN SMS Marketing</i>	43
3.2.8 <i>rmN eCommercePanel</i>	43
3.2.9 <i>rmN Alertas</i>	43
3.2.10 <i>rmN PermissionEmail</i>	44
3.2.11 <i>rmN eAnalysis</i>	45
3.3 Diseño Técnico del Sistema	46
3.4 Modelo de Base de Datos.....	50
3.5 Desarrollos JAVA destacados	51
3.5.1 <i>Búsqueda iterativa y creación de grupos</i>	51
3.5.2 <i>Generación de campañas y newsletters</i>	51
3.5.3 <i>Campañas y newsletters</i>	53
3.5.4 <i>Web y extranet</i>	53
3.5.5 <i>Productos y clientes</i>	54
3.5.6 <i>Alertas web, email y sms</i>	55
3.5.7 <i>Administración de rmN</i>	55
3.5.8 <i>Permission e-mail</i>	56
4. TRATAMIENTO DIGITAL DE LA INFORMACIÓN	57
4.1 Ensayos preliminares	57
4.1.1 <i>Definición del Problema</i>	58
4.1.2 <i>Métodos Lineales Analizados</i>	58
4.1.2.1 <i>Regresor lineal + umbral</i>	59
4.1.2.2 <i>Decisor basado en la regla del Perceptrón</i>	59
4.1.2.3 <i>Adaline con activación blanda</i>	60
4.2 Aplicación del Tratamiento Digital de la Información al mundo del Marketing....	63
5. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS	67
5.1 Conclusiones	67
5.2 Futuras líneas de desarrollo.....	69
6. REFERENCIAS Y OTRAS FUENTES	71
6.1 Referencias	71
6.2 OTRAS FUENTES	73
6.2.1 <i>Artículos</i>	73
6.2.2 <i>Publicaciones</i>	73
7. ANEXO 1. GESTIÓN DEL PROYECTO.....	75
Planificación del proyecto	75
Presupuesto	78
8. ANEXO 2. DISEÑO COMERCIAL DEL SISTEMA.....	79
9. ANEXO 3. EXTRACTOS DE CÓDIGO FUENTE.....	82
Detalle de programación Módulo Análisis	82
Clases Implementadas	85
Código Matlab.....	87

Índice de figuras

Ilustración 1: Ciclo de Vida de las soluciones CRM [4]	9
Ilustración 2: Valor Presente Neto por Cliente	11
Ilustración 3: Cuadrante Mágico de Gartner para CRM 2005	14
Ilustración 4: Pantallazo del módulo de Análisis de Siebel	15
Ilustración 5: Evolución de las soluciones CRM según Gartner 2008-2012	16
Ilustración 6: Pantallazo salesforce.com	17
Ilustración 7: Reparto del espacio de mercado entre las soluciones CRM. Forrester Wave 2010	18
Ilustración 8: Arquitectura Lógica JDBC	26
Ilustración 9: Arquitectura de conectividad JDBC	26
Ilustración 10: Ejemplo de una regresión lineal simple	33
Ilustración 11: Ejemplo de Red Neuronal	33
Ilustración 12: Perceptrón Monocapa con 2 Entradas	34
Ilustración 13: Entrada al Sistema	38
Ilustración 14: Menú Parametrizable de rmN	40
Ilustración 15: Flujo de Trabajo (Proceso) de E-mail Marketing en rmN v4.0.1	42
Ilustración 16: Modelo de Programación en Tres Capas	46
Ilustración 17: Arquitectura Lógica de rmN como parte de eCustomer Services	46
Ilustración 19: Maquina de Modelos Lineales	59
Ilustración 20: Resultados de la Aplicación de Adaline a los Datos de MERA	65
Ilustración 21: Diagrama Gantt Del Proyecto	77

Índice de tablas

Tabla. 1: Funcionalidades del Módulo rmN Administrador	39
Tabla. 2: Funcionalidades del Módulo rmN Contactos	41
Tabla. 3: Funcionalidades del Módulo rmN Campañas Online	41
Tabla. 4: Funcionalidades del Módulo rmN Newsletter	42
Tabla. 5: Funcionalidades del Módulo rmN SMSMkting	43
Tabla. 6: Funcionalidades del Módulo rmN eCommercePanel	43
Tabla. 7: Funcionalidades del Módulo rmN Alertas	44
Tabla. 8: Funcionalidades del Módulo rmN PermissionEmail	44
Tabla. 9: Funcionalidades del Módulo rmN eAnalysis	45
Tabla. 10: Requisitos Técnicos del Sistema	48
Tabla. 11: Compatibilidad de rmN	49
Tabla. 12: Datos Recogidos para el entrenamiento del Adaline	64
Tabla. 13: EDT (Estructura de Desglose de Tareas) del Proyecto	76
Tabla. 14: Matriz de Funcionalidades	80

ÍNDICE DE TABLAS

Capítulo 1

Introducción y objetivos

1.1 Contexto¹

La clave para alcanzar y mantener una posición de ventaja en un entorno altamente competitivo, caracterizado por la globalización y la apertura de mercados derivada de las recientes tendencias liberalizadoras, reside en la figura del cliente y la relación que la empresa mantenga con éste.

La empresa debe conocer quiénes son sus clientes, quiénes entre ellos son sus clientes más rentables, cómo se comportan y por qué, qué necesitan, qué desean, cómo y cuándo. En definitiva, la empresa debe conocer a sus clientes para poder poner en práctica aquellos mecanismos y estrategias de mercado que le permitan establecer una relación duradera y rentable con ellos, manteniendo en todo momento su fidelidad y valor.

CRM (Customer Relationship Management) es el proceso que posibilita reorientar los mecanismos estratégicos de la empresa desde la visión centrada en el producto hacia una perspectiva referida a la figura del cliente.

A través de este proceso, la empresa maximiza la información de la que dispone acerca de sus clientes con el fin de incrementar su conocimiento acerca de ellos y

¹ http://www.marketingycomercio.com/numero15/00jul6_crm.htm

construir a partir de tal conocimiento relaciones altamente rentables y duraderas con aquellos clientes en particular que sean de su interés.

Esto es, las soluciones CRM permiten incrementar los beneficios de la empresa a través de la adecuada gestión de las relaciones con sus clientes y el resto del mercado.

La implantación de una solución CRM consta de tres fases: la integración de las distintas fuentes de información de cliente en un repositorio común, el análisis de tal información y la acción (bajo la forma de definición y ejecución de campañas y acciones de marketing y comerciales concretas) a partir del conocimiento obtenido tras el análisis, e incluye la realimentación continua de cada nueva transacción con el cliente de modo que la empresa esté preparada para responder dinámicamente a la información y conocimiento asociado a tales interacciones.

Internet, la WWW y la telefonía móvil constituyen los ámbitos tecnológicos de mayor crecimiento en la actualidad. Conforme la empresa enriquece sus procesos de negocio mediante nuevos modelos de venta y canales de distribución e interacción con el cliente basados en la Web, en la telefonía móvil o en sistemas mixtos en los que se combinen ambos mundos (WAP), la solución CRM proporciona el marco de trabajo necesario para la gestión de las transacciones inherentes a tal interacción y su personalización e individualización conforme a las necesidades y requerimientos particulares del cliente, permitiendo de este modo combinar de un modo óptimo las oportunidades de expansión propias de Internet y la telefonía móvil con las necesidades y beneficios asociados a los sistemas CRM. [1]

Como puede verse a simple vista, esta introducción es eminentemente ‘marketiniana’ mas cercana a una visión de relaciones publicas que de una ingeniería de telecomunicaciones y la razón es clara, el proyecto ha sido preparado al amparo de una empresa, y como tal, si no la parte mas importante, al menos la mas diferente que he podido aprender es que en el mundo laboral, una buena idea, si no se consigue vender, deja de ser tan buena.

En la carrera hemos visto desde las distintas tecnologías el como abordar los proyectos, pero hemos dejado de lado el marco donde se insertan nuestras soluciones. Desde este punto de vista he decidido introducir en primer termino la necesidad que cubre mi proyecto, el target al que se dirige, mis usuarios, e intentar hacer ver lo primero de todo la utilidad de mi desarrollo, porque me ha resultado tan interesante y enriquecedora la consecución de este proyecto. A partir de este momento, después de presentar el documento de venta del CRM implementado en mi proyecto y la empresa que invirtió para que este proyecto viera la luz.

1.2 Objetivos y Motivación

La solución CRM persigue un propósito final: el incremento de los beneficios de la empresa a través de una mejor relación, basada en un mayor conocimiento, con el cliente. Esta proposición de partida se concreta en diversos aspectos que abarcan el amplio espectro de la estrategia y funcionamiento empresarial², los cuales se convirtieron en los objetivos de nuestro proyecto:

1. La mejora de la eficiencia de los procesos de relación con los clientes de la empresa, tanto los ya existentes como aquellos susceptibles de serlo en el futuro (clientes potenciales).
2. El incremento del conocimiento disponible en la empresa sobre tales clientes y, por tanto, del grado de diferenciación e individualización entre distintos clientes.³
3. La detección de nuevas oportunidades de marketing y venta derivadas del conocimiento adquirido sobre cada cliente (marketing one-to-one).
4. La mejor adecuación de las ofertas y servicios a las necesidades o deseos del cliente, consecuencia derivada directamente del mayor conocimiento sobre él adquirido por la empresa.
5. La reducción de los costes asociados a las campañas de venta y marketing. El conocimiento adquirido del cliente permite a la empresa personalizar sus campañas de modo que sólo aquellos segmentos de cliente susceptibles de responder a una promoción dada sean objeto de ella, reduciendo el coste asociado a campañas masivas de captación.
6. Todo ello es posible gracias al conocimiento personal que se va adquiriendo gracias al análisis basado en teoría de la información aplicada a cada acción realizada sobre los clientes.
7. El mercado Pyme reclama soluciones fáciles de usar e implementar, más que soluciones con muchas funcionalidades. Empresas de este tipo, a las que nos dirigimos les interesa adquirir soluciones asequibles y fáciles de utilizar.

En definitiva, un proyecto CRM puede ser muy beneficioso para la empresa, pero hay que prestar atención a una serie de requisitos antes de abordarlos, si se quiere tener éxito. En primer lugar hay que establecer los mecanismos adecuados para recopilar, ordenar e interrelacionar la información relativa a las relaciones con el Cliente, de manera que el departamento comercial vea este trabajo no como una amenaza, un control de su

² [2]

³ Universidad Politécnica Salesiana

<http://dspace.ups.edu.ec/bitstream/123456789/585/3/CAPITULO%20II.pdf>

actividad, sino como una herramienta pensada para ellos, que les ayude en su día a día, y para que las empresas no pierdan el know-how comercial. Una vez superada esta fase, puede realizarse la implantación de la solución elegida.

A nivel técnico, se quería abordar varios retos como fueron:

1. Desarrollo de un producto modular parametrizable. Dada la flexibilidad requerida, se veía clara la opción de utilizar programación orientada a objetos como JAVA.
2. Tecnología web. Para lograr estos puntos no se podía programar sólo con los conceptos aprendidos en la carrera, por tanto fue necesaria la investigación de los métodos de programación web disponibles con JAVA, su aprendizaje y utilización, como son JSP y Servlet.
3. Acceso a base de datos robusto. De igual forma que para web la investigación nos llevó a JSP y Servlet, para la base de datos decidimos optar por JDBC⁴.
4. Por último, quisimos dotar al sistema de una componente diferencial que ayudara a MERA a la toma de decisiones en la inversión de marketing sobre sus clientes. Como veremos a lo largo del proyecto, realizamos una investigación sobre los decisores que mejor se podían comportar en nuestro escenario, siempre dentro un margen de tiempo marcado por las entregas, que nos llevó a implementar un decisor basado en pesos calculado mediante el algoritmo de Adaline.

Todos estos retos se resolvieron con éxito en el marco del proyecto y la solución fue puesta en producción no sólo para MERA si no para un par de clientes más y sigue estando en el offering de Planet Media para soluciones CRM.

⁴ Java DataBase Connectivity, más conocida por sus siglas JDBC, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

1.3 Estructura de la memoria

En adelante, el cuerpo de la memoria se divide en 3 partes como son.

- a) Lo primero que abordaremos es el estado del arte actual y sobre todo el existente en el momento de realización del proyecto con mayor profundidad. Son dos los aspectos que se trataran, por uno, el contexto de negocio, es decir, el CRM y por otro la programación con JAVA.
- b) El segundo bloque está dedicado al producto en sí, el CRM que inicialmente fue un desarrollo a medida y que convertimos en producto comercial. Su nombre quedó como rmN. En este punto se comenta tanto desde el punto de vista comercial, como de arquitectura así como los aspectos de programación más relevantes en que consiste el producto. Se incluye un capítulo especial dedicado al completo a la solución de basada en regresión lineal para la decisión sobre la inversión en cliente y otro sobre la gestión del proyecto.
- c) Al final se incluyen las conclusiones y posibles mejoras sobre el sistema, que dado el tiempo transcurrido son muchas y de bastante calado.

Capítulo 2

Estado del Arte

2.1 CRM

2.1.1 Concepto de CRM⁵

Existen diferentes maneras de definir qué es el CRM (Customer Relationship Management):

Una estrategia centrada en el cliente, que busca un crecimiento en beneficios a través de proporcionar un mayor valor al cliente⁶.

Es la definición más simple que existe. Indica el fin de la estrategia, que lógicamente es ganar más dinero, y el medio a través del cual pretendemos conseguirlo: proporcionando al cliente un mayor valor. A primera vista es una definición interesante; sin embargo, nos faltan aún muchos elementos por introducir.

⁵ [4]

⁶ Mark Rieger, VP Marketing and Sales, Market Vision, Inc.

*Una estrategia de negocio que busca construir proactivamente un sesgo o preferencia por una organización con sus empleados, canales y clientes, que resulta en una mayor retención y un rendimiento económico superior.*⁷

En primer lugar, el objetivo de la experiencia no tiene por qué ser necesariamente el cliente final como tal; podemos encontrarnos estrategias de CRM destinadas a los empleados, a los canales, a los proveedores, etc. Por otro lado, esta última definición nos indica lo que queremos hacer: que el individuo al que se dirige la estrategia nos prefiera por encima de otras opciones. Y, por último, el fin, que no varía con respecto a la definición anterior: ganar más dinero.

*La base no es cuota de mercado, sino cuota de cliente. El marketing masivo desarrolla un producto y busca clientes para él. En cambio, el marketing 1 to 1 desarrolla un cliente y busca productos para él*⁸.

Los “inventores” de la filosofía cliente-céntrica no utilizan el término CRM porque lo ven excesivamente ligado al mundo de la informática; en su lugar, proponen Marketing One-to-One. En este caso, la definición va un paso más allá, e indica, de manera breve y brillante, dónde radica el cambio de paradigma y cómo se contrapone al marketing tradicional.

Alrededor del CRM aparecen modalidades de marketing estrechamente relacionadas, entre las que destacan el Permission Marketing o Marketing de Permiso.

Se trata de gestionar la relación con el cliente de tal manera que nos permita mantener o, mejor aún, ir escalando en el nivel de permiso que éste nos otorga. Deth Godin, ex-VP de Márketing de Yahoo!

Puede entenderse el CRM como el permanente intento de progresar en la relación con el cliente, intentando mantener con él una relación de mutua confianza que nos permita ofrecerle más y más productos o servicios, con la seguridad de que le serán útiles y no los percibirá como una invasión de su privacidad.

Además el concepto de CRM comprende la metodología, disciplina y tecnología (eminentemente materializada bajo la forma de sistemas software) que tiene por objeto automatizar y mejorar los procesos de negocio asociados a la gestión de la relación de la empresa con el cliente, principalmente en las áreas de venta, marketing, servicios de atención al cliente y soporte, con el fin último de incrementar los beneficios de la empresa mediante la optimización, personalización y diferenciación de dicha relación con el cliente.

La solución CRM combina una adquisición de información de los clientes de la empresa con la aplicación de una serie de tecnologías para la gestión de tal información y

⁷ Carlson Marketing Group

⁸ *Don Peppers, Martha Rogers The One to One Future*

su conversión en conocimiento de negocio (análisis estadístico, minería de datos, etc.) Ello proporciona, al mismo tiempo, una plataforma común para la comunicación e interacción con el cliente que minimice las limitaciones en la integración asociadas a la tradicional estructura organizativa que distingue las visiones de los departamentos de ventas, marketing y atención al cliente, facilitando una perspectiva homogénea de la figura del cliente y la gestión uniforme de su relación con la empresa desde cada punto de contacto.

2.1.2 Ciclo de vida de la solución CRM

En la economía tradicional, partimos de una masa indeterminada de clientes "grises", de los que sabemos muy poco, o por lo menos, menos de lo que querríamos saber. Visto así, resulta lógico que lo primero que intentemos hacer en una economía centrada en el cliente sea **IDENTIFICAR** a éstos, pintarlos de colores para poder reconocerlos, independientemente del medio por el que vengan, mediante una serie de métodos que veremos más adelante.

Tras ese paso inicial, nos plantearemos ser capaces de clasificar o **DIFERENCIAR** a estos clientes, mediante algún tipo de escala que nos revele su valor, tanto actual como potencial. Ahí surgirá la necesidad de realizar y analizar mediciones, sin las cuales no podríamos llevar a cabo esa tarea de una manera fiable.

La solución CRM es efectiva sólo si el conocimiento adquirido durante la etapa de análisis se materializa en acciones concretas durante la fase de **INTERACCIÓN** sobre los procesos de negocio, por lo que la revisión y modificación de dichos procesos para dar cabida a las conclusiones extraídas del análisis de la información de cliente (lo aprendido sobre las preferencias, necesidades y comportamiento del cliente) constituye la etapa que cierra el ciclo que estudiaremos en este proyecto de la solución CRM.

Ello se plasmará sobre el modelo de negocio de la empresa a través de alguno de los siguientes procedimientos:

- Identificación de actuaciones (campañas).
- Definición de criterios/mecanismos de actuación (acciones dentro de una campaña).
- Activación de acciones (interacción con los servicios y canales de atención al cliente).

Todo ello redunda en que sea en esta etapa del ciclo CRM donde se apliquen realmente las decisiones de carácter táctico y estratégico adoptadas: los procesos de negocio y las estructuras organizativas se refinan basándose en la mejor comprensión del comportamiento y necesidades del cliente adquiridas mediante el análisis de los datos recolectados previamente. La planificación de negocio y financiera se ve del mismo

modo revisada e integrada en todas aquellas actividades que impliquen un trato con el cliente, incluyendo entre tales los servicios de atención al cliente, marketing y ventas. La aplicación de tales medidas permitirá que se concreten los beneficios para la empresa de la implantación de la solución CRM. El resultado obtenido a través de las acciones adoptadas pasará a formar parte de la información recogida en el repositorio de datos y que será procesada en una próxima iteración, cerrando el ciclo de vida del sistema.

Dentro del sistema de CRM presentado en este proyecto (rmN), la fase de análisis continúa dentro de la parte de acción, haciendo de este un CRM ‘dinámico’, ya que podemos ir mejorando nuestro análisis en la medida que avanzamos con nuestras acciones. Dado que el tipo de empresa al que se dirige (alquiler de maquinaria para la construcción) no ‘está acostumbrado’ a este tipo de tecnología, la idea de análisis dinámico es la más acertada, ya que no tienen un Datawarehouse orientado a este tipo de relaciones, por tanto las primeras acciones serán un poco a ciegas y desde los resultados de las mismas podremos ir ajustando los datos de nuestro análisis. Por ello pensamos que es una buena solución para este proyecto.

El último paso, la **ADAPTACIÓN**, consiste en ser capaces de adaptar nuestros productos a esos clientes de los que ya sabemos más cosas, proporcionando el producto como a ellos les interesa, o, incluso, dándoles productos que no son fabricados por nosotros, pero que sabemos que les interesan, siempre y cuando mantengamos una cierta coherencia y orden en la mentalidad del cliente. Esta fase queda fuera del marco de este proyecto.

En la siguiente figura (Ilustración 1) vemos de forma esquemática el ciclo de vida de un CRM.

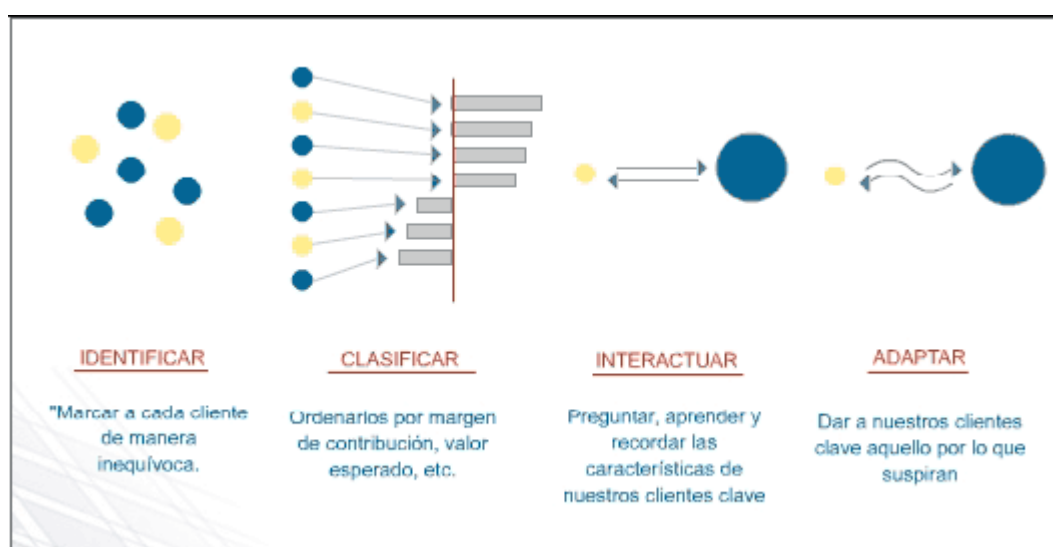


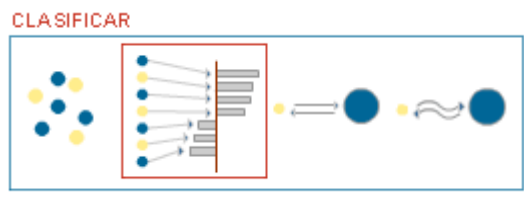
Ilustración 1: Ciclo de Vida de las soluciones CRM [4]

2.1.3 Clasificación de Clientes Dentro de las Soluciones CRM

Veamos con algo más de detalle la segunda etapa: cómo adscribir a esos clientes ya identificados un valor determinado. Ese valor debe asignarse con la misma minuciosidad que utilizamos, por ejemplo, para valorar una empresa o un proyecto.

Existe un variado abanico de técnicas que pueden utilizarse para calcular dicho valor. Entre ellas destacan:

- ☐ Valor de la vida útil del cliente
- ☐ Espectro de Mercado del cliente
- ☐ Otras métricas
- ☐ Minería de Datos



Dentro de las diferentes opciones que existen para evaluar los clients, dentro del marco del proyecto de rmN nos decantamos por utilizar uno método enclavado en el ultimo punto, Dataming que se explica con más detalle en el punto 2.3 y el capítulo 5.

2.1.3.1 Valor de la vida útil del cliente

El CLV o Valor de la vida útil del cliente (del inglés Customer Lifetime Value) es el método más comúnmente citado. Su cálculo responde a un proceso del tipo descuento de cash-flows, idéntico al utilizado para valorar una empresa o proyecto. Se trata, en primer lugar, de obtener los flujos reales, teniendo en cuenta no sólo lo que el cliente se gasta, sino todos los factores que puedan incidir en su valor.

Un cliente que atrae a muchos otros, por ejemplo, o que se atiende a sí mismo mediante autoservicio sin demandar atención de personal especializado, vale más que otro que no nos trae a nadie o requiere detallada atención. En el caso de un bar, el cliente que “nos llena el bar” y, además, pide algo normal nos supone un valor mayor que el que se sienta solo en un extremo de la barra y además pide un combinado que el camarero tarda diez minutos en preparar.

Finalmente, esos flujos obtenidos se descuentan a un tipo determinado, que será tanto mayor cuanto mayor sea la incertidumbre que tenemos sobre los parámetros estimados.

De una manera gráfica:

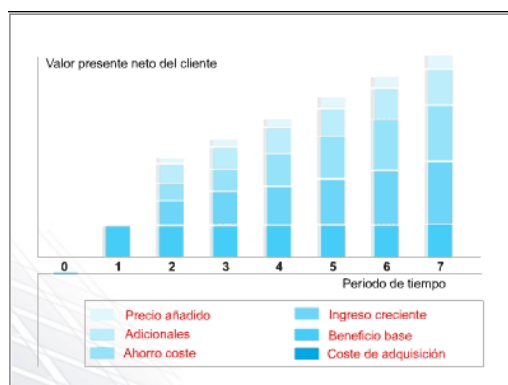


Ilustración 2: Valor Presente Neto por Cliente

El gráfico representa un ejemplo típico. Tras invertir una cantidad en captar al cliente (resultante de dividir, por ejemplo, el coste de la campaña entre el número de clientes que ha atraído), éste consume una cantidad determinada con una periodicidad determinada, que nos reporta su beneficio base. Sin embargo, el cliente puede incrementar ese consumo básico a medida que se fideliza, vía estrategias de desarrollo de clientes, de manera que tendríamos, además, un crecimiento en beneficios. Por otra parte, puede que, al conocerlo, nos resulte más barato darle servicio, con lo que tendríamos unos ahorros de costes operativos; que, además, nos traiga a un cierto número de clientes adicionales o recomendados; o, incluso, que esté dispuesto a pagar un precio algo superior por su deseo de seguir utilizando nuestros servicios, dada la mayor calidad o comodidad que representamos para él. Para calcularlo multiplicar el número esperado de visitas al bar por la media de importe gastado en cada visita:

- ☐ Deducir los costes de adquisición
- ☐ Deducir los costes de servicio a ese cliente
- ☐ Añadir el valor de los clientes que este cliente refiere
- ☐ Descontar la suma adecuadamente para el período de tiempo considerado

2.1.3.2 Espectro de Mercado del cliente

El Espectro de Mercado del cliente es la media inversa al CLV. Se trata de valorar cuánto podría llegar a valer un cliente para nosotros en el mejor de los casos, y averiguar cuánto de ese valor hemos capturado nosotros. Se llega así a una indicación de cuánto del gasto de ese cliente en la categoría somos capaces de capturar, que nos da una medida de hasta qué punto una estrategia de desarrollo de clientes nos podría reportar un beneficio. Para calcularlo, se divide el customer lifetime value que asignamos a ese cliente de acuerdo a nuestras prácticas actuales por el valor potencial que podría de hecho tener.

Si el valor estimado es de 1.000 y el potencial es de 10.000, nuestro share of customer sería del 10%.

Además, estudia cuál es la oportunidad para incrementar el customer lifetime value mediante la oferta al cliente de más cosas además de las que ya le ofrecemos.

Ejemplo

Un cliente puede tener domiciliada su nómina y sus recibos con nosotros, pero podemos ver que efectúa una transferencia automática todos los meses y también podemos saber o intuir que corresponde, por ejemplo, al pago de un crédito personal en otro banco. Esto nos permitiría calcular el porcentaje de ese cliente que tenemos y ofrecerle productos o servicios destinados a incrementar ese porcentaje.

2.1.3.3 Otras Métricas De Uso Habitual:

Existen otras variadas medidas, que hacen referencia a conceptos relacionados.

- ☐ *Repeat-customer conversion rate*. Porcentaje de consumidores que se convierten en repetitivos.
- ☐ *Customer churn rate*. Porcentaje de consumidores que no vuelven a comprar.
- ☐ *Repeat-customer revenue momentum*. Incremento en el ingreso trimestral derivado de los consumidores repetitivos.
- ☐ *Repeat-customer maintenance cost*. Costes operativos invertidos en cada cliente repetitivo, excepto los costes de marketing.

Algunas de estas medidas se han hecho muy populares en determinados sectores por necesidades puntuales como es el caso del churn rate en telefonía móvil. Los clientes cambiaban de proveedor de telefonía con tanta facilidad, que era preciso medirlo y valorarlo de cara a estimar el valor de las propias compañías. Evidentemente, un salto entre proveedores alto suele corresponderse con un producto percibido como de uso habitual o poco diferenciado. Otras medidas sirven a otros propósitos particulares, pero lo importante es percibir que el CRM tiene una serie de medidas rigurosas y calculables y que en ningún caso se trata de una alquimia o de una moda, sino de algo tangible y cuyos rendimientos se pueden calcular.

2.1.3.4 Minería de Datos

Otra manera de analizar la información de los clientes viene dada por la utilización de técnicas estadísticas conocidas.

- ☐ Clasificación
- ☐ Cubos OLAP
- ☐ Regresión
- ☐ Análisis de series temporales
- ☐ Análisis de cluster
- ☐ Asociaciones
- ☐ Análisis de secuencias
- ☐ Etc.,.

☐ **Clasificación:** La clasificación es, aparentemente, el método más sencillo para extraer conclusiones y consiste simplemente en la ordenación de listas mediante un criterio determinado. Sin embargo, la potencia del método, cuando se superponen diferentes criterios a subdivisiones progresivas, permite alcanzar conclusiones a veces muy interesantes.

☐ **Cubos OLAP:** el On Line Analytical Processing (OLAP) consiste en representar las tablas de doble entrada tradicionales en un soporte metodológico más complejo que permite cruzar más de dos variables, y subdividir la muestra entre las celdas resultantes de dividir en intervalos cada una de ellas. Se considera una herramienta del tipo “vistazo rápido”.

☐ **Regresión:** se trata de uno de los métodos más poderosos para identificar correlaciones y sus magnitudes, así como para diferenciar efectos aparentes de efectos reales. Sin embargo, resulta enormemente complicado aplicar la regresión de forma correcta, atendiendo al criterio de multicolinealidad, que podría invalidar nuestros resultados, y respetando cada uno de los supuestos iniciales.

☐ **Análisis de series temporales:** potentísimo método que permite observar el comportamiento de una variable en el tiempo y, después, desestacionalizarla en función de parámetros conocidos. Por ejemplo, eliminar el efecto en la serie del fin de semana, las vacaciones u otros efectos evidentes, puede permitirnos observar otras pautas que previamente se hallaban enmascaradas.

☐ **Análisis Cluster:** es una de las tecnologías más comúnmente citadas. Se trata de establecer una serie de ejes para las variables objeto de estudio y representar los puntos

en el espacio multidimensional creado. Permite identificar grupos homogéneos para hacerlos objeto de acciones específicas.

Debido a los conocimientos previos adquiridos durante la carrera, el método de clasificación de clientes elegido para desarrollar dentro de este proyecto, y por tanto, de aplicación en la plataforma rmN es el de **Regresión lineal**.

2.1.4 Espacio de Mercado de soluciones CRM

Como en casi todos los procesos de negocio de una empresa, existen multitud de soluciones de mercado que cubren las principales necesidades de los clientes.

Estas soluciones son parametrizables, tienden a estar verticalizadas por cada mercado y requieren de compra de licencias y de un proyecto de parametrización de la herramienta a la necesidad concreta del cliente particular.

De 2003 a 2008, las principales soluciones de CRM las ofrecían fabricantes como Oracle, Siebel (ahora comprada por Oracle), SAP, Peoplesoft y otras...como se puede apreciar en el siguiente Cuadrante Mágico de Gartner ⁹de 2005:

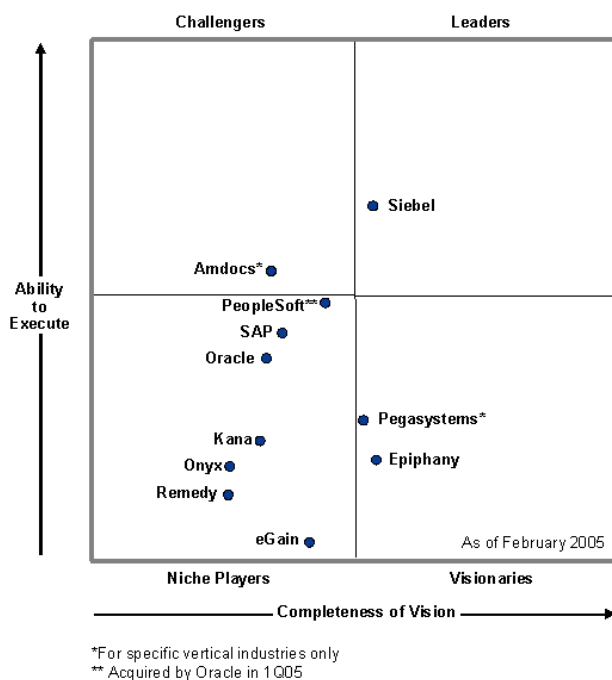


Ilustración 3: Cuadrante Mágico de Gartner para CRM 2005

⁹ Cuadrante Mágico de Gartner: es el nombre de la marca de una serie de informes de investigación de mercado publicados por Gartner, una compañía norteamericana de consulta e investigación. De acuerdo con Gartner, su Cuadrante Mágico trata de proveer un análisis cualitativo de un mercado y su dirección, madurez y principales actores.
http://en.wikipedia.org/wiki/Magic_Quadrant

Estas soluciones de mercado estaban diseñadas para grandes clientes con gran capacidad de inversión ya que es necesario una compra de licencias que suele ser cara, una compra de infraestructura suficiente, el proyecto y el servicio de evolución.

Como se puede ver en la siguiente figura (Ilustración 4), estas soluciones incluyen un alto grado de contenido de análisis gráfico, inicialmente algo pobre, pero que aportan gran valor al cliente.

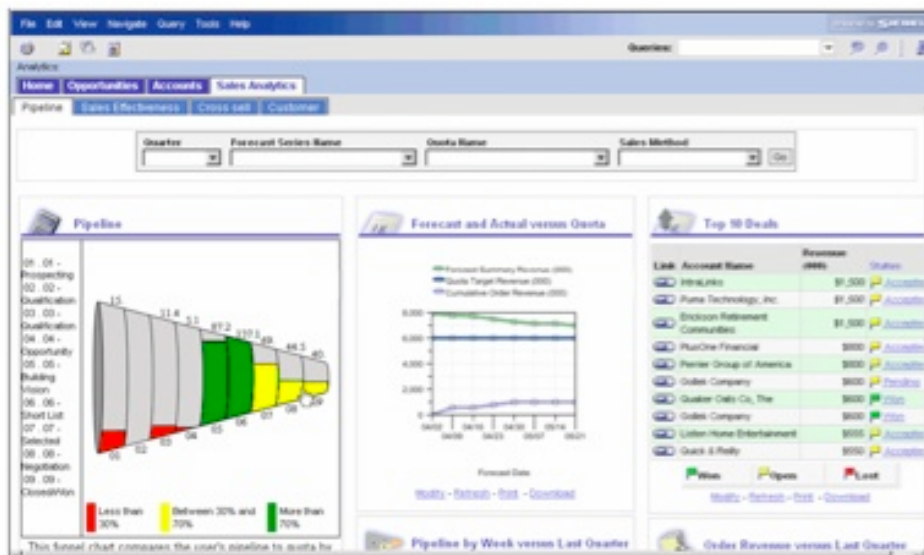


Ilustración 4: Pantallazo del módulo de Análisis de Siebel

Por esta razón, vimos en 2003 una buena idea invertir en crear a partir de un desarrollo a medida un producto CRM que permitiera a las PYMEs tener una gestión de clientes a un precio asequible.

Al pasar los años, más empresas vieron este nicho y ya por 2008 había soluciones que estaban cambiando el panorama con herramientas de fácil parametrización, sin coste alguno de inversión ni en Hardware ni en Software, creando así las primeras SaaS (Software as a Service¹⁰). Una de las más asentadas en salesforce.com, que ha pasado desde su irrupción en el cuadrante de Gartner en 2008 siendo el líder indiscutible en la actualidad, véase Ilustración 5.

¹⁰ Software as a service (SaaS, pronounced sæs or sas[1]), Software como servicio, se refiere a software consumido bajo demanda provisto por un proveedor de servicios de internet o de aplicaciones, normalmente en un modelo de pago por uso.
http://en.wikipedia.org/wiki/Software_as_a_service

Capítulo 2: Estado del Arte

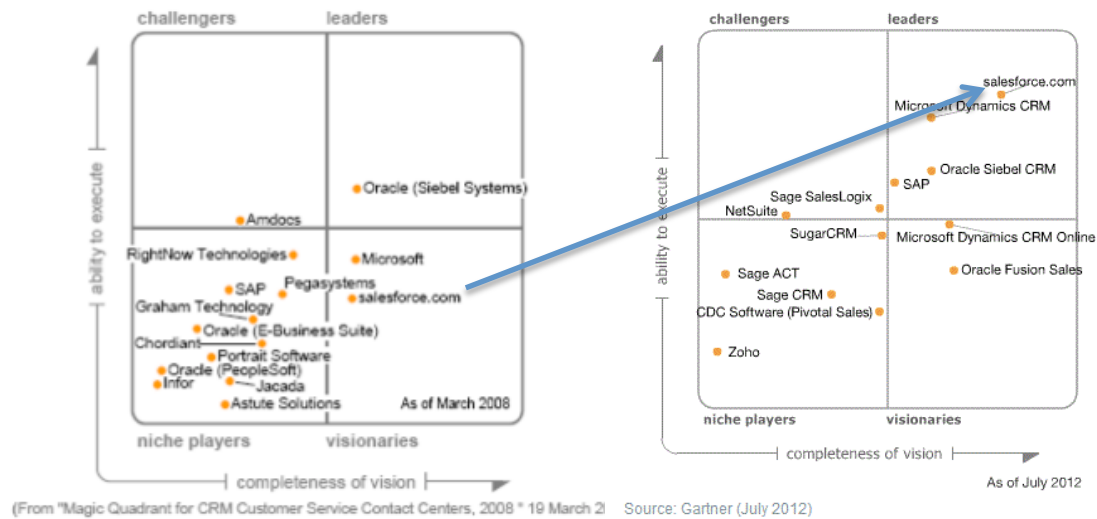


Ilustración 5: Evolución de las soluciones CRM según Gartner 2008-2012

Así sería una vista de la actual herramienta líder del sector, salesforce.com, donde vemos que la capacidad gráfica ha crecido enormemente.



Ilustración 6: Pantallazo salesforce.com

En la siguiente ilustración número 7 podemos ver como el reparto de clientes entre las principales soluciones en 2012 sigue manteniendo en una posición muy alta a los dos competidores de siempre y como irrumpen Microsoft y Salesforce.com. Este análisis de Forrester¹¹ apoya el anterior presentado de Gartner.

¹¹ Forrester Research es una compañía de investigación de mercado y tecnología independiente que provee consejo sobre el potencial impacto de la tecnología a sus cliente y el publico en general.

The Forrester Wave™: CRM Suites For Large Organizations, Q3 '12

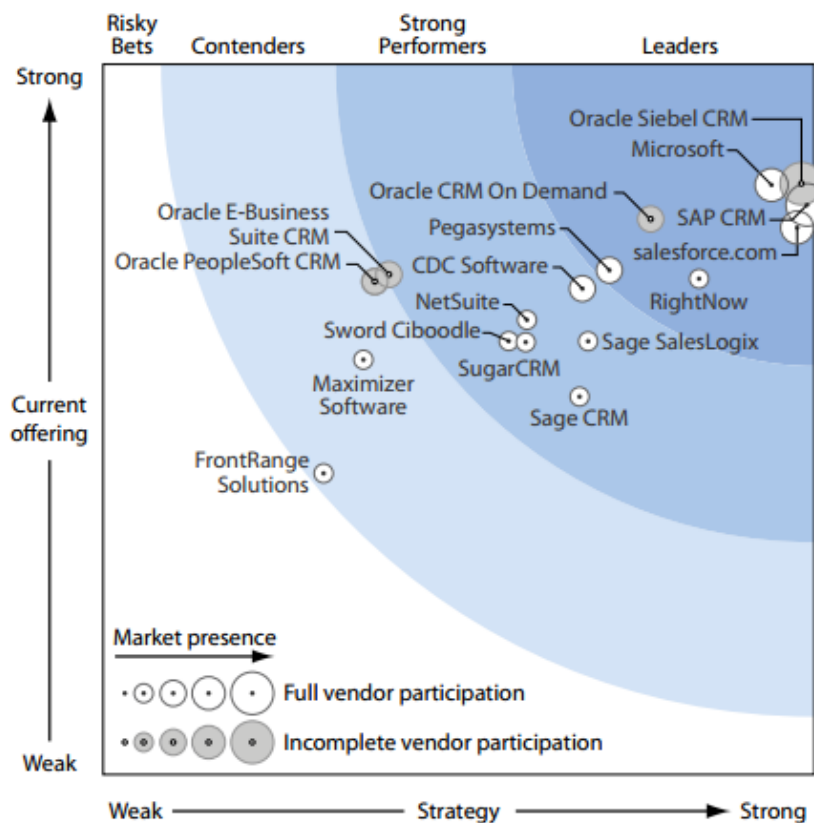


Ilustración 7: Reparto del espacio de mercado entre las soluciones CRM. Forrester Wave 2010

2.2 Tecnología Utilizada

Aunque el proyecto fue terminado el 2002 y hay muchas cosas que han avanzado desde entonces, todo lo aplicado sigue siendo válido, desde la primera decisión, utilizar JAVA como lenguaje, hasta los API utilizados.

JAVA sigue siendo el lenguaje de programación por excelencia por todas las características que ahora comentaremos. Es cierto que .NET ha cogido bastante ritmo, sigue siendo la segunda.

Incluso después de su compra por Oracle cerca de 2009, sigue siendo prácticamente 100% gratis y sigue teniendo un gran equipo de desarrolladores detrás que siguen sacando versiones y adaptándolo a las nuevas tendencias como es el mundo móvil, smartphones, tablets y ahora la computación en la nube o el big data¹², de hecho el

¹² "Big data" es un término aplicado a conjuntos de datos que superan la capacidad del software habitual para ser capturados, gestionados y procesados en un tiempo razonable. Los tamaños del "big data" se hallan

estándar utilizado para la computación distribuida más utilizado es Hadoop, desarrollado completamente en JAVA.

Las API más importantes que utilizamos, como son JDBC, JSP, Servlet y java mail siguen siendo utilizadas, de hecho, tanto JDBC como java mail siguen siendo el estándar.

Sobre JSP, la parte web, se ha evolucionado desde el HTML hacia el XHTML y de la misma forma se puede utilizar JSF en vez de JSP, pero se siguen creando páginas web dinámicas con JSP.

Para la parte servidora, donde nosotros utilizamos Servlet, existe una tecnología más actual que son los EJB, pero de igual forma siguen formas de funcionamiento diferente y cada una tiene sus pro y sus contras, y ambas siguen siendo utilizadas en 2013.

Algo que podría haber sido interesante explorar, sobre todo por el boom de los smartphones y tablets de los últimos años es hacer la aplicación 100% compatible con estos dispositivos utilizando HTML5 o con aplicaciones nativas, aunque, dado el número de usuarios (5) y el uso (sobre todo como herramienta de oficina) no habría sido visto como algo clave para el cliente.

Quizá sí que hubiera sido un punto sobre el que investigar todo lo relacionado con Big Data y la información desestructurada, y que el módulo de análisis pudiera tener en cuenta si una campaña tiene más aceptación lanzándola un día de lluvia o un día de huelga o con el estreno de una película.

Así como incluir módulos para campañas y gestión de las redes, de hecho, existe una rama del CRM de reciente creación que cubre este aspecto de forma independiente por su importancia, el Social CRM.

Con todo esto, el estado del arte actual ha variado un poco y si bien en 2002 “lo último” era pasar de cliente pesado a tecnología web y bases de datos relacionales, como fue este proyecto, ahora habría sido interesante haber incluido el concepto Social y las bases de datos desestructuradas, dos de los puntos que incluimos como proyectos futuros

2.2.1 Qué es Java

En un primer nivel, Java es un lenguaje de programación de propósito general, orientado a objetos, que fue introducido por Sun Microsystems en 1995, y diseñado en principio para el ambiente distribuido de Internet. Pero lo que hace de Java un concepto diferente es que, en un segundo nivel, es también un entorno para la ejecución de programas, englobado en la llamada máquina virtual de Java. Este entorno es un software que permite que las aplicaciones escritas en Java se ejecuten en cualquier ordenador, independientemente del sistema operativo y de la configuración de hardware utilizados.

constantemente en aumento. En 2012 se dimensionaba su tamaño en una docena de terabytes hasta varios petabytes de datos en un único data set.

2.2.2 ¿Por qué JAVA?

Vamos a exponer brevemente las principales cualidades que me han llevado a elegir este lenguaje de programación sobre otros que ofrece el mercado.

Universalidad. Aunque un programa interpretado no es en principio tan rápido como un programa equivalente compilado, las prestaciones de Java son sin embargo muchísimo mejores que las de cualquier lenguaje interpretado. Este hecho, junto con la sencillez de programación en Java (a la que nos referiremos luego) ha propiciado que se hayan escrito intérpretes de pequeño tamaño adaptados a prácticamente cualquier plataforma, desde mainframes y ordenadores personales (con cualquier sistema operativo: Windows, Macintosh OS, Unix,, etc.) hasta dispositivos electrónicos de bajo coste. Además, la universalidad de los byte codes hacen de Java el lenguaje idóneo para desarrollar aplicaciones para Internet. De hecho, la mayor parte de los navegadores (Google Chrome, Firefox, Internet Explorer) integran máquinas virtuales, y por tanto intérpretes de Java, lo que hace posible acceder automáticamente a los applets presentes en las páginas web. De nuevo la sencillez de Java hace que esta integración no reduzca en absoluto las prestaciones de los navegadores, permitiendo además la ejecución rápida y simultánea de gran cantidad de applets.

También se suele hacer referencia a la universalidad de Java con términos equivalentes como transportabilidad, o independencia de plataforma, pues para ejecutar un programa basta compilarlo una sola vez: a partir de entonces, se puede hacer correr en cualquier máquina que tenga implementado un intérprete de Java.

Además, las bibliotecas estándar de funciones y métodos de Java (definidas en su API, Application Programming Interface) facilitan la programación de multitud de acciones complejas (desarrollo de interfaces gráficas, multimedia, multitarea, interacción con bases de datos,, etc.). Ningún otro lenguaje (ni compilado ni interpretado) dispone como Java de una cantidad tan grande de funciones accesibles en cualquier plataforma sin necesidad de cambiar el código fuente.

Sencillez. Java es un lenguaje de gran facilidad de aprendizaje, pues en su concepción se eliminaron todos aquellos elementos que no se consideraron absolutamente necesarios. Por ejemplo, en comparación con otros lenguajes como C ó C++, es notable la ausencia de punteros, o lo que es lo mismo: es imposible hacer referencia de forma explícita a una posición de memoria; ello ahorra gran cantidad de tiempo a los programadores, dado que el comportamiento imprevisto de los punteros es una de las principales fuentes de errores en la ejecución de un programa. Por otra parte, el código escrito en Java es por lo general mucho más legible que el escrito en C ó C++.

Por otro lado, Java dispone de un mecanismo conocido como de "recogida de basura", el cual —usando la capacidad multitarea de Java— hace que, durante la ejecución de un programa, los objetos que ya no se utilizan se eliminen automáticamente de la memoria. Dicho mecanismo facilita enormemente el diseño de un programa y optimiza los recursos de la máquina que se esté usando para la ejecución del mismo (con

los lenguajes tradicionales, la eliminación de objetos y la consiguiente optimización de recursos debe planificarse cuidadosamente al idear el programa).

Orientación a objetos. Aunque C++ es también, como Java, un lenguaje orientado a objetos, la diferencia fundamental es que Java lo es desde su concepción, mientras que C++ se diseñó como un lenguaje compatible con C (que no es orientado a objetos): de este modo, un programa escrito en C++ puede ignorar la mayoría de las características de orientación a objetos de C++ y compilarse sin problemas en un compilador de C++. Sin embargo, un programador no puede obviar la orientación a objetos cuando escribe un programa en Java, y esto hace que las aplicaciones escritas en Java tengan interesantes ventajas.

Para entender qué es la orientación a objetos, recordemos que los lenguajes tradicionales no orientados a objetos, como Pascal ó C, están pensados para trabajar de forma secuencial y basan su funcionamiento en el concepto de procedimiento o función. La tarea principal del programa se divide en funciones o tareas más pequeñas, a menudo muy interrelacionadas, por lo que es difícil modificar una función sin tener que revisar el resto del programa: por tanto, también es difícil reutilizar o actualizar los programas ya escritos.

En el caso de los lenguajes orientados a objetos, el concepto clave no es el de función, sino el de objeto. Un objeto es un elemento de programación, autocontenido y reutilizable, y que podríamos definir como la representación en un programa de un concepto, representación que está formada por un conjunto de variables (los datos) y un conjunto de métodos (o instrucciones para manejar los datos). Por ejemplo, en una aplicación de Bibliotecas un objeto puede ser un código de barras, que contiene datos (p. ej. el número mismo del código de barras) e instrucciones para manejarlos (p. ej. el método para calcular el dígito de control). Los objetos, además, poseen la capacidad de enviarse mensajes entre sí durante la ejecución de un programa.

La "encapsulación" de variables y métodos en un objeto tiene claras ventajas:

1. Cada objeto puede ser modificado y mantenido por separado.
2. Se pueden mantener en un objeto métodos y variables que no son accesibles desde fuera de él, lo que evita multitud de posibilidades de error en el momento de confeccionar un programa. Esta característica se llama "ocultamiento de la información".
3. Es posible reutilizar porciones de programa ya escritas: en el ejemplo anterior, un programador puede reutilizar el objeto "código de barras" para escribir software para diversos tipos de transacciones, como préstamo, devolución, reserva, etc. sin necesidad de reescribir cada vez los comandos para calcular el dígito de control, dado que el programa se encargará de hacerlo cada vez que se use el objeto.
4. El estándar JavaBeans de Java proporciona un entorno visual de programación que facilita y simplifica la programación de objetos. De hecho, es posible escribir applets sencillos en Java sin necesidad de escribir una sola línea de código.
5. Seguridad. En general, se considera que un lenguaje es tanto más seguro cuanto menor es la posibilidad de que errores en la programación, o diseños

malintencionados de programas (virus), causen daños en el sistema. La extrema seguridad de Java se establece a tres niveles:

- a. Nivel de seguridad dado por las características del lenguaje, tales como la ausencia de punteros (que evita cualquier error de asignación de memoria) o el "ocultamiento de la información" propio de la programación orientada a objetos, por recordar dos ejemplos ya mencionados.
 - b. Nivel de seguridad dado por el diseño de la VM¹³. La VM de Java posee un verificador de los byte codes, que antes de ejecutarlos analiza su formato comprobando que no existen punteros en ellos, que se accede a los recursos del sistema a través de objetos de Java, etc.
 - c. Nivel de seguridad dado por la API de Java. El conjunto de métodos y clases que estamos obligados a utilizar cuando programamos en Java para acceder a los recursos del sistema, está definido por la API, y constituye la última barrera defensiva. El diseño de dichos métodos y clases hace que éstos realicen múltiples verificaciones cuando son invocados, de modo que se dificultan los errores (voluntarios o involuntarios).
6. Otro elemento constitutivo de la VM es el cargador de clases. Una clase es una categoría de objetos utilizados en un programa; cuando se ejecuta un programa en Java, éste llama a determinadas clases a través del cargador de clases. Estas clases pueden provenir de tres lugares distintos, en donde residen en forma de ficheros: del ordenador local, de la red de área local a la que pueda estar conectado el ordenador cliente, o de Internet. En función de la procedencia de las clases, se efectúan una serie de comprobaciones diferentes y el gestor de seguridad de la VM prohíbe los accesos peligrosos.
7. Adaptación a redes (y en particular a Internet). Java irrumpió en el mercado para potenciar la interactividad en Internet, y todos hemos "ejecutado" algún applet alguna vez, (pequeños programas en Java que se cargan junto con una página web desde un servidor, y que son ejecutados -por la VM del navegador del cliente- como una parte de la página web). Además de las ventajas que supone su ejecución local, los applets disponen de una significativa riqueza de recursos y son capaces de realizar tareas muy complejas a pesar de su reducido tamaño. Una de las explicaciones de esta sorprendente capacidad es el hecho de que los applets se sirven del propio código del navegador en cuya VM se ejecutan, utilizándolo para tareas tales como presentación gráfica o comunicaciones. Sin embargo, el acceso a las funciones del navegador es totalmente automático y transparente para el programador, que debe limitarse a invocar ciertas funciones de la API de Java; estas invocaciones, interpretadas por el navegador, dan origen a acciones muy complejas. Esta observación es muy importante cuando se discute del rendimiento de Java, pues todas estas acciones se realizan en la

¹³ Máquina Virtual de Java, del inglés Virtual Machine

máquina que está ejecutando el applet, y la rapidez de ejecución de las mismas no depende de que Java sea un lenguaje semiinterpretado (o semicompilado).

Entre las tareas básicas más comunes que suelen realizar los applets, se encuentran:

- a. visualizar animaciones en la ventana del navegador;
- b. reproducir sonidos;
- c. establecer comunicaciones con el servidor del que procede el applet (p. ej. para cargar desde él ficheros de cualquier naturaleza);
- d. crear interfaces gráficas con los elementos habituales de los entornos de ventanas (como Macintosh o Windows):
 - i. menús desplegables,
 - ii. botones,
 - iii. áreas de texto,
 - iv. barras de desplazamiento, etc.;
- e. pedir datos al usuario para procesarlos (gestionando eventos como pulsaciones de teclas o acciones con el ratón); etc.

No obstante, y como acabamos de decir, es posible programar applets para realizar tareas de enorme complejidad.

Después de enumerar, describir y analizar las principales cualidades que definen el lenguaje de programación Java, procederemos a focalizar dichas cualidades en los beneficios directos derivados de las características de Java.

Rapidez de desarrollo y mejora del software. El hecho de que Java sea un lenguaje orientado a objetos desde su concepción tiene, entre otras muchas consecuencias, la de que es fácil reutilizar el código de programación, como ya hemos comentado anteriormente; y por tanto los desarrollos de una aplicación serán más rápidos, pues es más rápido reutilizar objetos y sus componentes que rescribir el código desde el principio. Además, una vez que el código de un objeto es estable, la reutilización de ese objeto replica ese código fiable en cualquier parte en toda la aplicación, lo que reduce el proceso de depuración (debugging).

Además, como hay menor posibilidad de errores en la programación, como también hemos visto al describir los niveles de seguridad de Java, resulta que, como se ha dicho en alguna ocasión: "al programador en Java no le queda más remedio que escribir código robusto y fiable", o lo que es lo mismo: en el desarrollo de una aplicación se necesitará emplear menos tiempo (y dinero) en los procesos de depuración y de perfeccionamiento o reescritura del programa.

Otra consecuencia de la orientación a objetos de Java es que hace que el software escrito en Java sea modular: como el objeto es la entidad clave en la programación, cada uno puede ser modificado y mantenido por separado. Esta característica, ha sido básica a la hora de seleccionar Java como lenguaje de programación, dada la sencillez con la que podemos adaptar un lenguaje orientado a objetos al modelo de trabajo en capas que impera actualmente en el mercado.

Además, en Java no existe el concepto de fichero ejecutable: un programa no es más que un conjunto de ficheros compilados para la VM (llamados "módulos"), que no es

necesario "enlazar" en un único ejecutable como ocurría en el caso de los lenguajes compilados. Esto significa que pueden realizarse modificaciones sobre cada uno de los "módulos" sin necesidad de tener que recompilar y enlazar todos ellos: basta compilar sólo los "módulos" afectados.

En el caso de un sistema de gestión de Bibliotecas, esta modularidad implica que sus distintos módulos (en el sentido tradicional de este término: módulo de adquisiciones, módulo de control de publicaciones periódicas, etc.) pueden compartir una serie de applets que realicen las mismas funciones (como pueden ser los que proporcionan la interfaz gráfica, p. ej.), lo que incrementa la facilidad de desarrollo y mantenimiento del software.

Insistimos de nuevo en que la universalidad de Java equivale a que las aplicaciones escritas en Java son transportables de una plataforma a otra: pueden funcionar en cualquier máquina y bajo cualquier sistema operativo sin que sea precisa ninguna modificación, siempre que en dicha plataforma se haya instalado un intérprete o VM (máquina virtual) de Java: recuérdese que se han implementado VMs para casi todas las plataformas, incluyendo las más populares como Unix, MacOS, IBM OS/2, Microsoft Windows Server, ó Solaris. Pues bien, esta independencia total de la máquina elimina la necesidad de desarrollar y mantener múltiples versiones del código fuente. En particular, cuando se deseaba implementar una nueva funcionalidad de una aplicación escrita en un lenguaje compilado, podía ser necesario reescribirla en múltiples ocasiones para adecuarla a cada plataforma en que se utilizaba la aplicación. En el caso de Java, basta añadir el código una sola vez, en un solo lugar. Este enfoque es más eficiente, económico y directo que soportar diferentes paquetes de cliente para cada tipo de plataforma.

Seguridad, fiabilidad y eficiencia. Las características de Java como lenguaje redundan en una ejecución segura del código, de manera que es posible construir módulos de software capaces de detectar intentos de acceso a recursos privilegiados del sistema. Esa capacidad es importante, sobre todo a la hora de emplear Java en redes de ordenadores inseguras como Internet (cuando se cargan applets). La supremacía de Java sobre los lenguajes interpretados tradicionales (muy populares en Internet debido a su flexibilidad para evaluar cadenas dinámicas de caracteres, lo que permite manejar formularios, por ejemplo) es rotunda, pues éstos presentan graves deficiencias en ese sentido.

Para resumir, recordemos que la semicompilación de Java proporciona resultados aceptables, y más si la VM incorpora un compilador just-in-time. Podemos afirmar que la sencillez de las construcciones de que consta Java permiten obtener un código de tamaño reducido, por lo que la penalización en que se incurre (con respecto a la utilización de un lenguaje compilado) por la necesidad de interpretarlo se reduce considerablemente, y las diferencias en la rapidez de ejecución serán inapreciables por el usuario.

Una vez analizadas las principales cualidades y beneficios que nos ofrece Java sobre el resto de lenguajes de programación, pasaremos a definir dos API's específicos que han sido claves para el desarrollo del CRM para la empresa MERA.

2.2.3 Bases de datos distribuidas (JDBC)

Dentro de este apartado vamos a abordar el siguiente contenido:

- Descripción del API JDBC
- Arquitectura de JDBC
- Progreso a través de partners
- Apoyo de la industria
- Ventajas de la tecnología JDBC
- Características clave
- Documentos relacionados

El API de JDBC es un estándar desarrollado para la conexión entre programas Java y una amplia gama de bases de datos, que pueden residir en servidores independientes. El API JDBC abstrae los procesos de base de datos proporcionando una capa que aísla las llamadas Java de dichos procesos. La tecnología JDBC permite utilizar el lenguaje de programación Java para explotar el principio "programa una vez, utilízalo siempre" en todos los desarrollos que requieran acceso a los datos de la empresa.

2.2.3.1 Descripción del API JDBC

Se considera el primer producto estándar de Java con DBMS, creado y ofrecido por primera vez en marzo de 1996.

Crea una interfaz con un nivel de programación que le permite comunicarse con las bases de datos mediante un concepto similar al de componentes ODBC, el cual se ha convertido en el estándar que se utiliza en computadoras personales o en redes locales.

El estándar de JDBC está basado en un nivel de interfaz con instrucciones SQL X/Open, que es básicamente lo mismo que en ODBC.

Las clases de objetos para iniciar la transacción con la base de datos, están escritas completamente en Java, lo cual permite mantener la seguridad, robustez y portabilidad de este ambiente.

El puente JDBC-ODBC manipula la traducción de llamadas JDBC a aquellas que puedan ser entendidas por el cliente ODBC a un nivel de lenguaje C.

El API JDBC permite realizar tres funciones:

- Establecer una conexión con una BBDD o acceder a cualquier fuente de datos tabulada.
- Enviar sentencias SQL
- Procesar los resultados

2.2.3.2 Arquitectura de JDBC

El API JDBC contiene dos sistemas importantes de interfaces: el primero es el API JDBC para los desarrolladores de aplicaciones, y el segundo esta pensado para los

programadores de drivers y, por ello, llega a un nivel más bajo. Los drivers generados con la tecnología JDBC pertenecen a una de las cuatro posibles categorías que veremos en las ilustraciones que se muestran a continuación. Aplicaciones y applets pueden acceder a bases de datos vía API JDBC usando drivers basados en la tecnología Java JDBC pura, según vemos en esta figura:

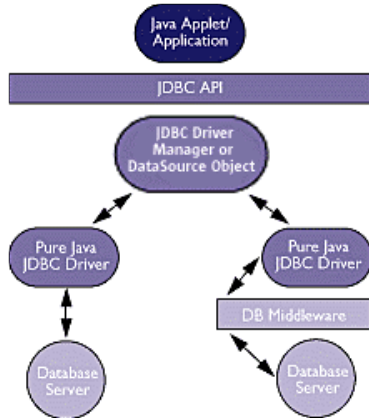


Ilustración 8: Arquitectura Lógica JDBC

En el lado izquierdo de la Ilustración 8 se muestra el acceso directo a la base de datos a través de driver puro de Java, lo que se denomina categoría 4. Este tipo de driver convierte las llamadas JDBC en el protocolo de red usado directamente por el servidor de BBDD, permitiendo una llamada directa desde el cliente al servidor y proporcionando una solución práctica para el acceso desde Intranet.

En el lado derecho de la Ilustración 8 corresponde con la categoría 4: Driver puro de Java para base de datos que usen middleware. Este tipo de driver convierte las llamadas JDBC en llamadas bajo el protocolo del proveedor del middleware, el cual será traducido al protocolo de BBDD a través de un servidor middleware. El middleware puede proporcionar conectividad a diversas bases de datos.

El siguiente gráfico ilustra la conectividad de JDBC usando drivers ODBC y librerías existentes del cliente de base de datos.

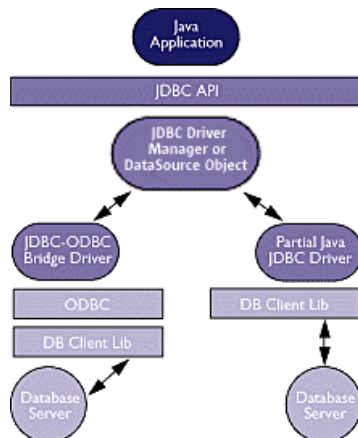


Ilustración 9: Arquitectura de conectividad JDBC

A la izquierda en la Ilustración 9 encontramos la categoría 1: Puente JDBC-ODBC más driver ODBC.

Esta combinación proporciona el acceso de JDBC a través de drivers ODBC. El código ejecutable ODBC - y en muchos casos, el código del cliente de la base de datos - debe estar instalado en cada máquina del cliente que utilice un puente JDBC-ODBC. SUN proporciona un driver puente de JDBC-ODBC, que es apropiado para el uso en desarrollos experimentales y para las situaciones en las cuales no hay otro driver disponible.

Por último, la denominada categoría 2 se puede ver en el lado derecho de la Ilustración 9: un API originario de la tecnología Java.

Este tipo de drivers convierten llamadas JDBC en llamadas al API del cliente de base de datos: MySQL, Oracle, Sybase, Informix,, etc. Destaca que, como ocurre en el driver puente, este tipo de driver requiere una aportación de código preinstalado en el cliente.

2.2.3.3 Apoyo del Mercado

Las compañías líderes en bases de datos, middleware y aplicaciones han introducido la tecnología JDBC en muchos nuevos productos. Esto asegura que los clientes puedan crear aplicaciones Java portables, pudiendo elegir entre una amplia gama de productos competitivos para encontrar la solución que más se ajuste a sus necesidades.

2.2.3.4 Ventajas de la tecnología de JDBC

- **Conservación del Legacy:** Con la tecnología JDBC las empresas no se limitan a ninguna arquitectura propietaria y pueden continuar con el uso de sus bases de datos instaladas y acceder fácilmente a la información, incluso con distintas tecnologías de bases de datos.
- **Desarrollo Simplificado para la Empresa:** La combinación del API Java y del API JDBC hace que el desarrollo de aplicaciones sea fácil y económico. JDBC oculta la complejidad de muchas tareas de acceso a los datos, haciendo gran parte del “trabajo pesado” del programador detrás de la interfaz. JDBC es intuitivo, sencillo de usar y mantener.
- **Configuración cero para sistemas en red:** Con el API JDBC no se requiere ninguna configuración en el lado del cliente. Con un driver escrito en el lenguaje de programación Java, toda la información necesitada para hacer una conexión es definida totalmente por la URL de JDBC o por un objeto de DataSource registrado con un Java Naming and Directory Interface (JNDI) que nombra el servicio. La configuración cero para los clientes apoya en el paradigma de servicios en red y centraliza el mantenimiento del software.

2.2.3.5 Características Clave

- Acceso completo a Metadata: El API JDBC proporciona el acceso al metadata que permite el desarrollo de sofisticadas aplicaciones que necesitan conocer la configuración y las capacidades subyacentes de una conexión específica con la base de datos.
- Sin Instalación: Un driver basado en la tecnología JDBC no requiere ninguna instalación especial; se descarga automáticamente como parte del applet que hace las llamadas JDBC.
- Conexión a la base de datos identificada por una URL: La tecnología JDBC explota las ventajas del estándar de nomenclatura URL de Internet para identificar conexiones de la base de datos. El API JDBC incluye una manera mejor de identificar y conectar con una fuente de datos, usando un objeto de DataSource, que hace el código aún más portable y más fácil mantener.
- Además de esta ventaja importante, los objetos de DataSource pueden proporcionar la conexión a un pooling y a transacciones distribuidas, esenciales para poder realizar determinadas actuaciones sobre la base de datos de la empresa. Esta funcionalidad se presenta transparente al programador.
- Incluido en la plataforma Java: Como pieza del núcleo de la plataforma Java 2, el API JDBC está disponible dondequiera que esta esté. Esto significa que se pueden escribir módulos de acceso a base de datos una vez y utilizarlos para conectarte desde cualquier aplicación. El API JDBC esta incluido en ambas plataformas de Java, la Standard Edition (J2SE) y la Enterprise Edition (J2EE), proporcionando la funcionalidad cliente - servidor obteniendo así una ventaja de cara al mercado.

2.2.3.6 Conectividad de Bases de Datos de Java (JDBC)

Se considera el primer producto estándar de Java con DBMS, creado y ofrecido por primera vez en marzo de 1996.

Crea una interfaz con un nivel de programación que le permite comunicarse con las bases de datos mediante un concepto similar al de componentes ODBC, el cual se ha convertido en el estándar que se utiliza en computadoras personales o en redes locales.

El estándar de JDBC está basado en un nivel de interfaz con instrucciones SQL X/Open, que es básicamente lo mismo que en ODBC.

Las clases de objetos para iniciar la transacción con la base de datos, están escritas completamente en Java, lo cual permite mantener la seguridad, robustez y portabilidad de este ambiente.

El puente JDBC-ODBC manipula la traducción de llamadas JDBC a aquellas que puedan ser entendidas por el cliente ODBC a un nivel de lenguaje C.

2.2.4 Java e Internet: jsp y servlet

Los Servlets son la respuesta de la tecnología Java a la programación CGI. Son programas que se ejecutan en un servidor Web y construyen páginas Web. Construir páginas Web dinámicamente es útil (y esta muy extendido) por una serie de razones:

- Una página web creada por un servlet puede ser modelada por datos recogidos del usuario. Un ejemplo podrían ser las páginas de resultados de los motores de búsqueda que recogen del usuario el dato sobre el que realizar la búsqueda y generan una página web con los resultados encontrados. Otro ejemplo son los programas que procesan pedidos desde sites de comercio electrónico.

- Los datos de una página web cambian frecuentemente. Por ejemplo, un informe sobre el tiempo o páginas de cabeceras de noticias podrían construir la página dinámicamente, devolviendo una página previamente construida y luego actualizándola.

- Las páginas Web que usan información desde bases de datos corporativas u otras fuentes. Por ejemplo, se podrían utilizar servlet para hacer una página Web en una tienda on-line que liste los precios actuales y el número de artículos en stock.

Ventajas de los Servlets sobre el CGI "Tradicional":

- Los Servlets Java son más eficientes, fáciles de usar, más potentes y portables que otras muchas tecnologías del tipo CGI.

- Eficiencia. Con un CGI tradicional, se arranca un nuevo proceso para cada solicitud HTTP. Si el programa CGI hace una operación relativamente rápida, la sobrecarga del proceso de arrancada puede dominar el tiempo de ejecución. Con los Servlets, la máquina Virtual Java permanece arrancada, y cada petición es manejada por un thread Java de peso ligero, no un proceso pesado del sistema operativo. De forma similar, en CGI tradicional, si hay N peticiones simultáneas para el mismo programa CGI, el código de este problema se cargará N veces en memoria. Sin embargo, con los Servlets, hay N threads pero sólo una copia de la clase Servlet.

- Los Servlet también tienen más alternativas que los programas normales CGI para optimizaciones como los cachés de cálculos previos, mantener abiertas las conexiones de bases de datos, etc.

- Lenguaje ampliamente extendido. Los Servlets tienen una gran infraestructura para análisis automático y decodificación de datos de formularios HTML, leer y seleccionar cabeceras HTTP, manejar cookies, seguimiento de sesiones, y muchas otras utilidades.

- Potencia. Los Servlets Java nos permiten realizar muchas operaciones difíciles o imposibles con un CGI normal. Los servlets pueden hablar directamente con el servidor Web. Esto simplifica las operaciones que se necesitan para buscar imágenes y otros datos almacenados. Los Servlets también pueden compartir los datos entre ellos, lo que puede aportar una gran utilidad, como por ejemplo almacenes de conexiones a bases de datos fáciles de implementar. También pueden mantener información de solicitud en solicitud, simplificando cosas como seguimiento de sesión y el caché de cálculos anteriores.

- Portable. Los Servlets están escritos en Java y siguen un API bien estandarizado. En consecuencia, los servlets escritos en el servidor I-Planet Enterprise, se pueden

ejecutar sin modificarse en Apache, Microsoft IIS, o WebStar. Además los Servlets están soportados directamente o mediante plug-in en la mayoría de los servidores Web.

- Económico. Hay un número de servidores Web gratuitos o de muy bajo coste que son buenos para el uso "personal" o el uso en sites Web de bajo nivel. Sin embargo, con la excepción de Apache/Tomcat, que es gratuito, la mayoría de los servidores Web comerciales son relativamente caros. Para este proyecto, por razones económicas, la empresa ha optado por la combinación de Apache como servidor web y Tomcat como servidor de aplicaciones.

2.2.4.1 ¿Qué es JSP?

Java Server Pages (JSP) es una tecnología que nos permite mezclar HTML estático con HTML generado dinámicamente. Muchas páginas Web que están construidas con programas CGI son casi estáticas, con la parte dinámica limitada a muy pocas localizaciones. Pero muchas variaciones CGI, incluyendo los servlets, hacen que generemos la página completa mediante nuestro programa, incluso aunque la mayoría de ella sea siempre lo mismo. JSP nos permite crear dos partes de forma separada. Aquí tenemos un ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>Welcome to Our Store</TITLE></HEAD>
<BODY>
<H1>Welcome to Our Store</H1>
<SMALL>Welcome,
<!-- User name is "New User" for first-time visitors -->
<% out.println(Utils.getUserNameFromCookie(request)); %>
To access your account settings, click
<A HREF="Account-Settings.html">here.</A></SMALL>
<P>
Regular HTML for all the rest of the on-line store's Web page.
</BODY></HTML>
```

2.2.4.2 ¿Cuáles son las Ventajas de JSP?

JSP frente a Active Server Pages (ASP). ASP es una tecnología similar de Microsoft. Las ventajas de JSP están duplicadas. Primero, la parte dinámica está escrita en Java, no en Visual Basic, otro lenguaje específico de MS, por eso es mucho más poderosa y fácil de usar. Segundo, es portable a otros sistemas operativos y servidores Web.

JSP frente a los Servlets. JSP no nos da nada que no pudiéramos en principio hacer con un servlet.

Pero es mucho más conveniente escribir (y modificar!) HTML normal que tener que hacer un billón de sentencias println que generen HTML. Además, separando el formato

del contenido podemos poner diferentes personas en diferentes tareas: nuestros expertos en diseño de páginas Web pueden construir el HTML, dejando espacio para que nuestros programadores de servlets inserten el contenido dinámico.

JSP frente a Server-Side Includes (SSI). SSI es una tecnología ampliamente soportada que incluye piezas definidas externamente dentro de una página Web estática. JSP es mejor porque nos permite usar servlets en vez de un programa separado para generar las partes dinámicas.

Además, SSI, realmente está diseñado para inclusiones sencillas, no para programas "reales" que usen formularios de datos, hagan conexiones a bases de datos, etc.

JSP frente a Java Script. JavaScript puede generar HTML dinámicamente en el cliente. Esta es una capacidad útil, pero sólo maneja situaciones donde la información dinámica está basada en el entorno del cliente. Con la excepción de las cookies, el HTTP y el envío de formularios no están disponibles con Java Script. Y, como se ejecuta en el cliente, JavaScript no puede acceder a los recursos en el lado del servidor, como bases de datos, catálogos, información de precios, etc.

2.2.4.3 ¿Qué son los JSP?

Las JavaServer Pages (JSP) nos permiten separar la parte dinámica de nuestras páginas Web del HTML estático. Simplemente escribimos el HTML regular de la forma normal, usando cualquier herramienta de construcción de páginas Web que usemos normalmente. Encerramos el código de las partes dinámicas en unas etiquetas especiales, la mayoría de las cuales empiezan con "<%" y terminan con "%>".

Normalmente daremos a nuestro fichero una extensión .jsp, y normalmente lo instalaremos en el mismo sitio que una página Web normal. Aunque lo que escribamos frecuentemente se parezca a un fichero HTML normal en vez de un servlet, detrás de la escena, la página JSP se convierte en un servlet normal, donde el HTML estático simplemente se imprime en el stream de salida estándar asociado con el método service del servlet. Esto normalmente sólo se hace la primera vez que se solicita la página, y los desarrolladores pueden solicitar la página ellos mismos cuando la instalan si quieren estar seguros de que el primer usuario real no tenga un retardo momentáneo cuando la página JSP sea traducida a un servlet y el servlet sea compilado y cargado. Observa también, que muchos servidores Web nos permiten definir alias para que una URL que parece apuntar a un fichero HTML realmente apunte a un servlet o a una página JSP.

Además del HTML normal, hay tres tipos de construcciones JSP que embeberemos en una página: elementos de script, directivas y acciones. Los elementos de script nos permiten especificar código Java que se convertirá en parte del servlet resultante, las directivas nos permiten controlar la estructura general del servlet, y las acciones nos permiten especificar componentes que deberían ser usados, y de otro modo controlar el comportamiento del motor JSP. Para simplificar los elementos de script, tenemos acceso a un número de variables predefinidas como request del fragmento de código anterior.

2.3 Tratamiento Digital De La Información¹⁴

En investigación aplicada es muy común encontrar situaciones en las que debemos estimar o predecir el comportamiento de una variable criterio en función de una o varias variables predictoras. Cuando el criterio es una variable cuantitativa se suele hablar de problemas de predicción o estimación, mientras que cuando es una variable cualitativa/categorial se habla entonces de problemas de clasificación.

Tradicionalmente la solución a estos problemas se ha llevado a cabo desde la óptica de modelos estadísticos de regresión: regresión simple o múltiple para problemas de predicción; análisis discriminante o modelos de regresión logística para problemas de clasificación. Casi todas estas técnicas descansan en el llamado modelo de regresión general lineal, pudiendo ser conceptualizadas a su vez como casos particulares del análisis de correlación canónica (ver p.a. Knapp, 1978; Allison, Gorman y Primavera, 1993).

En estadística la regresión lineal o ajuste lineal es un método matemático que modela la relación entre una variable dependiente Y_{calc} , las variables independientes x_n a través de los coeficientes w_n . Este modelo puede ser expresado como:

Ecuación 1: Modelo de Regresión Línea:

$$w_0 + w_1 \times x_1 + w_2 \times x_2 + \dots + w_n \times x_n = Y_{calc}$$

Y_{calc} : variable dependiente, explicada o regresando.

x_1, x_2, \dots, x_n : variables explicativas, independientes o regresores.

w_1, w_2, \dots, w_n : parámetros o coeficientes, miden la influencia que las variables explicativas tienen sobre el regresando.

¹⁴ [13]

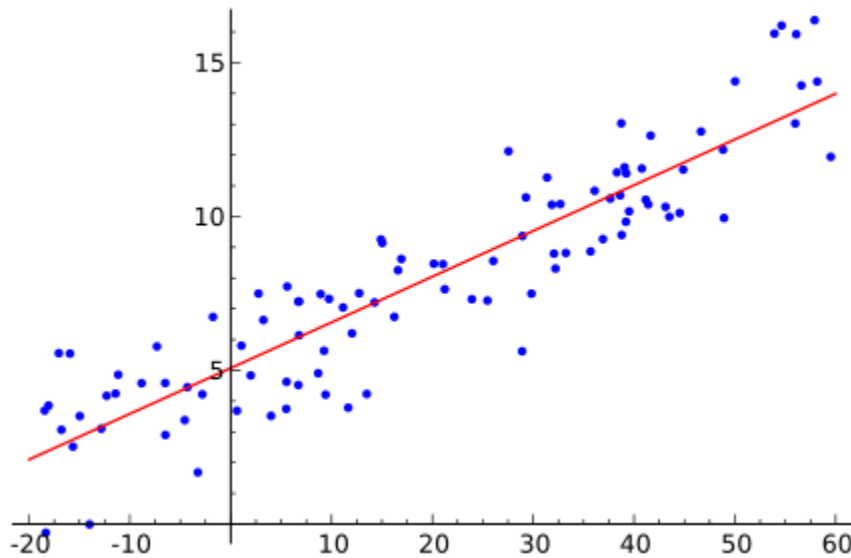


Ilustración 10: Ejemplo de una regresión lineal simple

A partir de 1986 (Rumelhart y McClelland, 1986; McClelland y Rumelhart, 1986) las llamadas redes neuronales o modelos conexionistas han ido progresivamente utilizándose como herramientas de predicción y clasificación.

Las redes de neuronas artificiales (denominadas habitualmente como RNA o en inglés como: "ANN") son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida. En inteligencia artificial es frecuente referirse a ellas como redes de neuronas o redes neuronales y se pueden representar como la figura anexa.

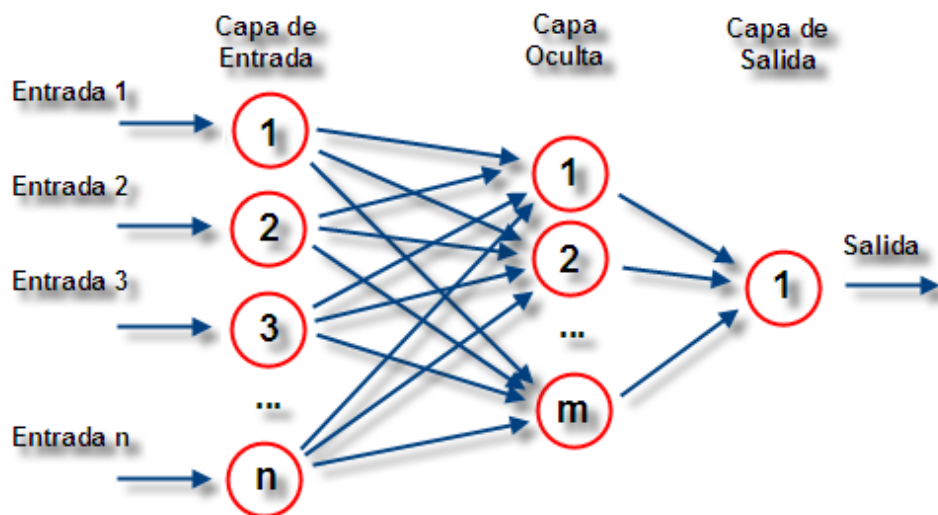


Ilustración 11: Ejemplo de Red Neuronal

Las redes neuronales consisten en una simulación de las propiedades observadas en los sistemas neuronales biológicos a través de modelos matemáticos recreados mediante mecanismos artificiales (como un circuito integrado, un ordenador o un conjunto de

válvulas). El objetivo es conseguir que las máquinas den respuestas similares a las que es capaz de dar el cerebro que se caracterizan por su generalización y su robustez.

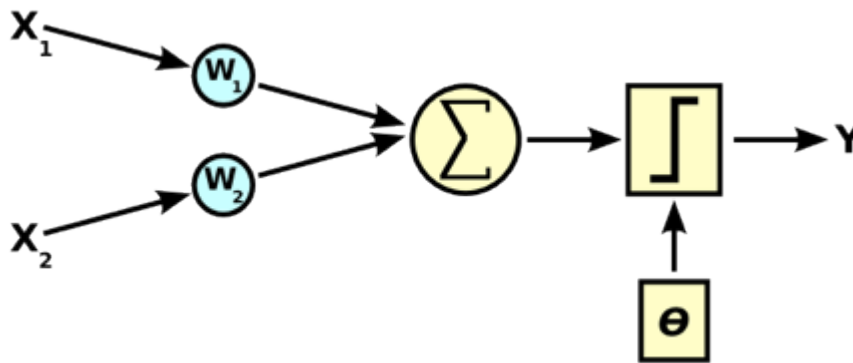


Ilustración 12: Perceptrón Monocapa con 2 Entradas

El objetivo del trabajo de [13] era contrastar sistemáticamente el rendimiento de redes neurales con modelos estadísticos convencionales.

De sus resultados se extrae que únicamente en la tarea de predicción cuantitativa y bajo condiciones idóneas de aplicabilidad, el procedimiento clásico de regresión obtuvo mejores resultados que las redes neurales. En el resto de condiciones redes neurales y modelos de regresión múltiple rinden por igual en este tipo de tarea.

Sin embargo en todo tipo de tareas de clasificación (binaria o no) las redes neurales obtuvieron resultados mejores en todas las condiciones que los modelos estadísticos convencionales.

Por tanto, y dado que nuestro problema dentro del CRM es la clasificación binaria, esto es, definir dos grupos de clientes, aquellos en los que la inversión podrá traer más frutos y aquellos donde no merece la pena invertir.

Dentro del campo de las redes neuronales aplicada a estructuras lineales, vamos a estudiar dos mecanismos, el Perceptrón y el Adaline.

El Perceptrón Simple (Rosenblatt, 1959), consta de una red con una capa de salida de n neuronas y otra de salida con m neuronas.

Utiliza señales binarias, tanto de entrada como de salida de las neuronas y su función de activación es de tipo signo.

Como regla de propagación se usa la suma ponderada del producto escalar por el vector de entrada, debiendo superarse un cierto umbral:

Ecuación 2: Modelo del Perceptrón

$$Decisión = \begin{cases} si, & Y_{calc} = 1 \\ no, & Y_{calc} = -1 \end{cases}$$

Adaline Generalmente se compone de una sola capa de m neuronas (por tanto m valores de salida) con n entradas con las siguientes características:

- Las n entradas representan un vector X de entrada..

- Por cada neurona, existe un vector w_i de pesos sinápticos que indican la fuerza de conexión entre los valores de entrada y la neurona. En la práctica representan la ponderación de cada entrada sobre la neurona.
- Una constante w_0 .
- La salida Y_{calc} de la neurona se representa por la función de activación, que se define como

Ecuación 3: Función de Activación Adaline

$$Y_{calc} = \sum_{i=1}^n w_i \times x_i + w_0$$

2.4 Conclusiones Del Estado Del Arte

Hemos visto, una vez analizado el estado del arte, que un CRM es una poderosa herramienta de marketing y vista su evolución, no nos confundimos en lanzar al mercado una herramienta web que pudiera cubrir el segmento de las pequeñas y medianas empresas, aunque el progreso de las redes sociales seguro que hubiera abierto un mundo de reflexión y trabajo hacia los social CRM.

Hemos visto que la tecnología elegida para el desarrollo, JAVA, sigue siendo el estándar de programación más extendido y que los componentes de acceso a base de datos o de publicación web siguen siendo validos, si bien, el avance de las tecnologías nos empujaría ahora a explorar otros estándares de programación que analizar.

Por último, el tratamiento digital de la información está cada vez más de moda, ya hemos comentado el llamado big data que nos proporcionaría millones de variables, de información que incluir en nuestros decisores para ayudar más en la toma de decisiones.

En conclusión, bajo mi humilde opinión el proyecto, en su año de finalización, estaba en la cresta de la ola en cuanto a las ideas y tecnologías escogidas, ahora mismo ha bajado, pero sigue en la ola y creo que ha sido un complemento perfecto para una carrera de ingeniería.

Capítulo 3

Diseño y Desarrollo del Sistema

3.1 Introducción

El diseño del producto y su desarrollo se ha dividido en 5 apartados:

- a) Diseño Comercial del Sistema (ver Anexo 2)
- b) Diseño Funcional del Sistema
- c) Diseño Técnico del sistema
- d) Diseño del Modelo de Datos. BBDD
- e) Desarrollos destacados

Al ser un producto comercial desarrollado dentro de una empresa para el uso de un cliente no es posible detallar todo el desarrollo realizado durante el proyecto, por esa razón, sólo se hace mención a ciertos desarrollos destacados por su interés técnico objeto de este proyecto y en acuerdo con todas las partes.

3.2 Diseño Funcional del Sistema

3.2.1 Caso de estudio

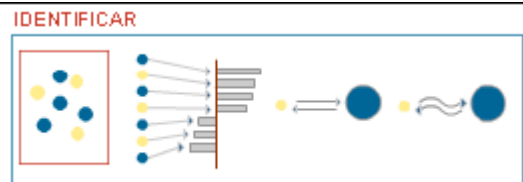
Nos encontramos frente a frente en un caso claro de aplicación de CRM que se sigue de la introducción planteada en el primer punto.

Una pyme dedicada al alquiler de maquinaria pesada MERA, detrás de la filial de Caterpillar en España, Finanzauto, quiere posicionarse dentro del emergente mundo 'www' lanzada por la creciente competencia en el sector (Ausa, CASE, Imcoinsa, PEYSER, Hertz) y que invertían parte de su presupuesto en su imagen corporativa on-line.

Dentro de MERA, como era lógico, el departamento marketing y relaciones públicas era nuestro cliente, ya que desde allí es donde más se puede beneficiar el global de la empresa.

Desde una primera idea de disponer de un Web para abrir negocio a través de la red, se fue dibujando un claro interés hacia una solución mas completa. Fue entonces cuando se propuso la implantación de una solución CRM.

La principal idea de la implantación era poder medir la capacidad de penetración que se conseguía a través de la Web, pero descubrieron dentro de rmN una herramienta mucho mas útil ya que desde ella, además de analizar el funcionamiento de la Web tendrían una integración completa de su Datawarehouse para lanzar sobre ella de manera sencilla una serie de acciones de marketing, campañas, newsletter, smsmktng, etc. y poder analizar los resultados de dichas campañas desde múltiples puntos de vista, entre ellos uno que incluye y compara la penetración de Web con las campañas personalizadas. Para ello era básico completar la primera fase del ciclo de vida del CRM, la **IDENTIFICACIÓN**, resuelta gracias al login dentro de la extranet y del uso de elementos de comunicación con el servidor para las campañas y Newsletter. De esta manera se podía ir creando un perfil de usuario y así ajustar cada vez más las acciones de marketing, haciendo que su efectividad aumentara sensiblemente gracias al modelo de análisis dinámico que incluye rmN.



En ese momento la aplicación aun no estaba creada, existían versiones anteriores que no tenían toda la funcionalidad que MERA necesitaba, por ello se opto por crear una nueva versión de la aplicación, pero partiendo de cero, llegando incluso a cambiar de tecnología para ello.

En el documento de requisitos recogido en sucesivas reuniones se fueron definiendo todas las funcionalidades que debía de incluir su CRM y nosotros, buscando que este proyecto a medida nos sirviera para lanzar un producto que fuera adaptable a cualquier cliente tanto como escalable para futuras versiones, dividimos la aplicación en los siguientes módulos que explicaremos con detalle en los siguientes puntos:

- Clientes
- Análisis
- Alertas
- Sms-marketing
- Campañas
- Newsletters
- Alquileres
- Permission.

3.2.2 Modelo Funcional

Como anticipaba en el punto anterior, el modelo funcional de rmN esta dividido en módulos. Cada módulo tiene una serie de funcionalidades propia y únicas que van añadiendo utilidades al CRM completando así todas posibilidades, de esta manera, cada cliente puede seleccionar solamente aquellos módulos que realmente son necesarios para su negocio.

Dentro de la idea modular de rmN se ha pretendido que sea lo mas usable para el cliente, proveyendo accesos rápidos desde unos módulos a otros centrando la información a una selección. Se pretende que en fases posteriores se puedan configurar dichos accesos¹⁵.

Solamente el módulo de clientes, que es el corazón de la aplicación, es imprescindible en todas las instancias de rmN. De igual modo, cada módulo se ha intentado modularizar a su vez en funcionalidades mas simple, confiriendo así una gran flexibilidad a la solución.

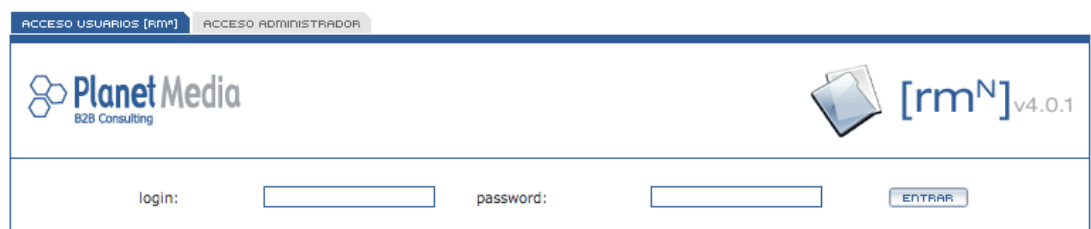


Ilustración 13: Entrada al Sistema

¹⁵ Ver desarrollos posteriores

Vamos a definir todos los módulos de rmN y añadiremos una lista con todas las funcionalidades de cada módulo.

3.2.3 rmN Administrador

Módulos	Funcionalidades	Definición
RmN Administrador		
	Generador de usuarios rmN¹⁶	Creación de usuarios de la aplicación
	Generador de usuarios rmN avanzado	Asignación de permisos
	Configurador alertas	Configuración de alertas
	Importador de plantillas	Importar plantillas para utilizar en Email, campañas y Newsletter

Tabla. 1: Funcionalidades del Módulo rmN Administrador

Este módulo se definió como plataforma de configuración de rmN. Desde aquí, con un usuario (*ver política de usuarios* en 5.2 Futuras líneas) administrador se pueden generar nuevos usuarios de rmN, asignarles permisos, modificarlos, etc. Inicialmente la política de usuarios se definió en un principio de una manera compleja y totalmente configurable, pero por usabilidad finamente se optó por dos tipos de usuarios, el administrador de la misma y el usuario final de la aplicación.

Dentro de la configuración desarrollada para este proyecto el usuario administrador podía configurar todas las alertas (*ver módulo de alertas*) activarlas, definir los mensajes, el modo de recepción (mail, aplicación, sms). También es desde este módulo desde donde se actualizan las plantillas usadas en los módulos de newsletter, campañas y envío de email.

En la siguiente Ilustración 15 se recoge el aspecto general de la aplicación donde se muestran los paneles lateral y superior que acompañan la navegación durante toda la solución, el cuál se construyó de forma que fuera totalmente parametrizable.

¹⁶ Para la definición de las funcionalidades, aparecerá en negrita aquellas funcionalidades que sean obligatorias. El único módulo obligatorio es el de Contactos

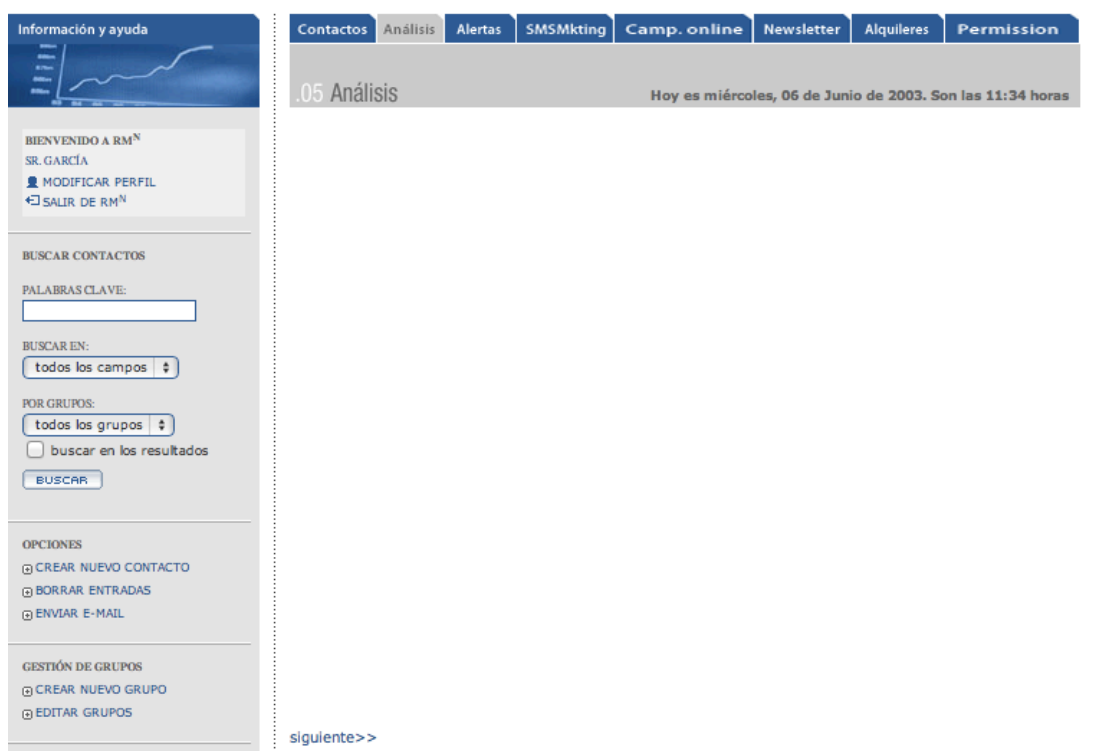


Ilustración 14: Menú Parametrizable de rmN

3.2.4 rmN Contactos

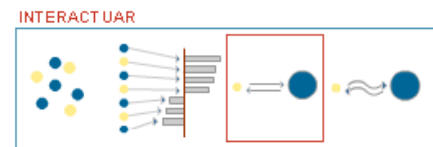
Este es el módulo central de rmN, su corazón. De este módulo se alimentan el resto de módulos ya que una campaña se tiene que enviar a alguien y un análisis sin un cliente al que referirse es inútil. Con este módulo se podrán dar de alta los cliente, hacer búsquedas, crear grupos en base a tantos criterios como se quieran anidar¹⁷, crear grupos en base a los resultados de las campañas, enviar email con plantillas corporativas y tener un acceso rápido al resto de módulos para ver la información de un cliente en concreto, alquileres de ese cliente, análisis, alertas, etc. Es el único módulo completamente indispensable.

Módulos	Funcionalidades	Definición
RmN Contactos		
	Panel de contactos	Crear usuarios, listado, accesos
	QuickEmail generator	Para enviar emails a los contactos
	QuickEmail advanced generator	Posibilidad de aplicar plantillas

¹⁷ Solución interesante que se analizara en el punto 3.4.Búsqueda

	SmartAccess QuickSMS	Acceso a envío de sms
	SmartAccess Alertas	Acceso a las alertas individuales
	SmartAccess eAnalysis	Acceso al análisis individual
	Buscador básico de contactos	Para cortar los usuarios por datos
	Buscador avanzado de contactos	Para cortar los usuarios por análisis
	Generador básico de grupos	Para crear grupos por datos
	Generador avanzado de grupos (eAnalysis)	Para crear grupos a partir de los datos del análisis
	SmartAccess eCommercePanel	Para acceder al panel de alquileres

Tabla. 2: Funcionalidades del Módulo rmN Contactos



3.2.5 rmN Campañas

Módulo creado para poder realizar campañas online. Dentro del ciclo de vida de un CRM, este sería uno de los módulos donde se gesta la interactividad con el cliente. Desde aquí podremos crear una campaña a partir de una plantilla¹⁸, incluyendo productos, vínculos con la web para que puedan establecerse hitos a la hora de analizar su penetración¹⁹. Dentro del menú de campañas se pueden hacer búsquedas para listar solo las campañas que interesen en momentos concretos.

En la figura 15 se ha incluido el proceso de marketing mediante correo electrónico que se automatizó con rmN en este módulo.

Módulos	Funcionalidades	Definición
rmN Campañas online		
	Generador de campañas básico	Para crear campañas y asignarlas a grupos
	Generador de campañas avanzado	Se pueden aplicar distintas plantillas y tener vista previa
	Buscador de campañas	Para listar campañas concretas
	SmartAccess eAnalysis	Acceso directo al análisis de la campaña

Tabla. 3: Funcionalidades del Módulo rmN Campañas Online

¹⁸ Solución interesante que se analizara en el punto 3.4. Plantillas

¹⁹ Solución interesante que se analizara en el punto 3.4. Analisis Campañas

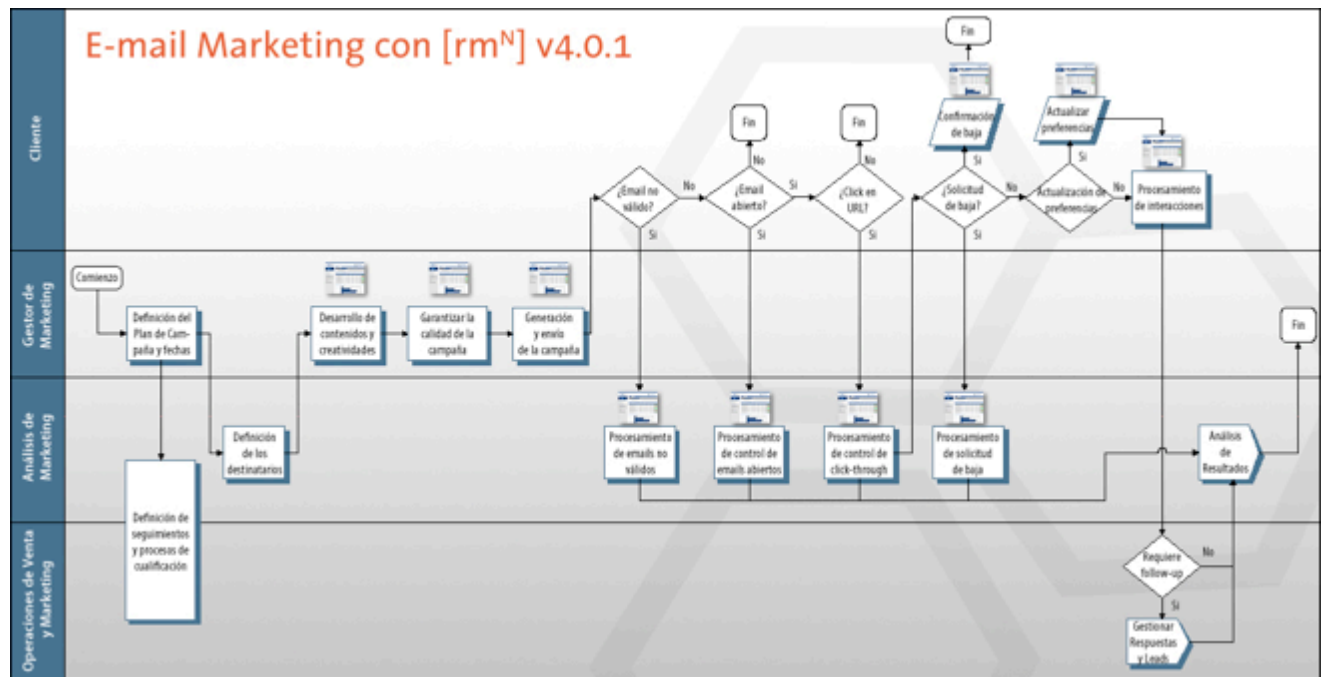


Ilustración 15: Flujo de Trabajo (Proceso) de E-mail Marketing en rmN v4.0.1

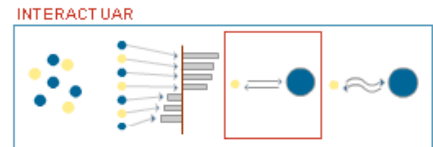
3.2.6 rmN Newsletter

Este módulo, funcionalmente es idéntico al anterior, la diferencia estriba en el uso y la interpretación de los datos, ya que una newsletter es mas informativa que comercial, y no suele ser muy específica. Por ello en este módulo se podrán añadir noticias a los envíos.

Módulos	Funcionalidades	Definición
rmN Newsletter		
	Generador de newsletter básico	Para crear campañas y asignarlas a grupos
	Generador de newsletter avanzado	Se pueden aplicar distintas plantillas y tener vista previa
	Buscador de newsletter	Para listar newsletter concretas
	SmartAccess eAnalysis	Acceso directo al análisis de la newsletter

Tabla. 4: Funcionalidades del Módulo rmN Newsletter

3.2.7 rmN SMS Marketing



Con este módulo abrimos un nuevo canal para la interacción con el cliente. Con los tan de moda sms, desde este módulo se pueden lanzar campañas personalizadas a tus contactos y realizar un seguimiento del resultado..

Módulos	Funcionalidades	Definición
rmN SMSMkting		
	Generador de campañas básico	Para crear campañas y asignarlas a grupos
	Seguidor de campañas	Para tener información del estado de la campaña
	Buscador de campañas	Para listar campañas concretas
	QuickSMS	Envío de sms discretos

Tabla. 5: Funcionalidades del Módulo rmN SMSMkting

3.2.8 rmN eCommercePanel

Este es el módulo que nos ayudara a gestionar el comercio a través de la web. A este módulo irán llegando los pedidos de los clientes y se podrá llevar el ciclo de venta a través de los estados de los pedidos.

Módulos	Funcionalidades	Definición
rmN eCommercePanel		
	Visualización de pedidos	Listado básico de pedidos con su estado
	Gestión de pedidos básica	Se puede ver el estado de los pedidos, los ítems seleccionados...
	Gestión avanzada de pedidos	Enlazada con el ERP de la empresa
	Buscador de pedidos	Para listar campañas concretas

Tabla. 6: Funcionalidades del Módulo rmN eCommercePanel

3.2.9 rmN Alertas

Este módulo, que puede ser configurado desde el módulo Administrador es el encargado de avisarnos de cualquier evento que se produzca en rmN²⁰. La entrada de un alquiler, alcanzar diez aperturas en una campaña, el envío de una campaña (útil si la

²⁰ Solución interesante que se analizara en el punto 3.5.Alertas

plataforma la utilizan varios usuarios) pueden ser avisos que lleguen por email, como entrada en el listado de alertas o incluso por sms.

Módulos	Funcionalidades	Definición
rmN Alertas		
	Panel de Alertas	Listado básico de alertas con su estado
	AlertTicker	Información adicional sobre las alertas
	AlertPopUp	Aviso PopUp de alertas.

Tabla. 7: Funcionalidades del Módulo rmN Alertas

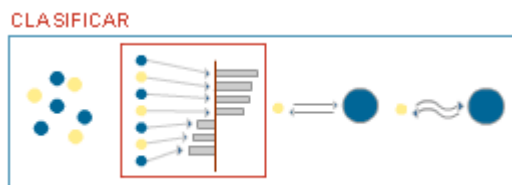
3.2.10 rmN PermissionEmail

Es ilegal emprender campañas de marketing sobre usuarios que no han dado su permiso expreso, o que no sean clientes de la empresa en cuestión. Pero sí que existen traspaso, cesión, intercambio, venta, etc. de bases de datos entre empresas, como manera de abrir nuevo negocio. Es por ello, y también para poder unificar distintos tipos de base de datos heterogéneos existentes en la misma empresa, que se ha ideado este módulo. Desde él, gracias a la interfaz que desarrollamos, se pueden conectar al cualquier base de datos, seleccionar la tabla y el campo, y crear un ‘campaña’ para presentar la empresa a nuevos clientes, o simplemente para incorporarlos a la base de datos de rmN.

Módulos	Funcionalidades	Definición
rmN PermissionEmail		
	Panel de Permission	Listado básico de permission con su estado
	Universal Relational DB Connector	Un conector universal para acceder a cualquier base de datos
	Generador paquetes de permission	Para crear los permission
	Envío permission	Envío de permission
	SmartAccess eAnalysis	Acceso rápido al análisis de Permission

Tabla. 8: Funcionalidades del Módulo rmN PermissionEmail

3.2.11 rmN eAnalysis



Desde el punto de vista práctico, este es el módulo más importante de rmN. Es aquí donde se generan y se analizan todos los informes y graficas que son posibles gracias a la plataforma rmN. Para este caso en concreto los puntos de análisis son las campañas, las newsletter, los permission y la extranet, y con todo se hace un perfil de los clientes. Sobre todo ello, a parte de mostrar los datos sin mas, se le añade un componente propio de rmN, una estimación del tipo de cliente, es decir, la 3ª parte del ciclo de un CRM, la **CLASIFICACIÓN**, malo, regular, bueno...a través de formulas obtenidas por medio del Tratamiento digital de la información²¹. Los puntos que se han analizado para este caso, por ser los mas sencillos son las aperturas o entradas y las selecciones²².

Módulos	Funcionalidades	Definición
rmN eAnalysis		
	Análisis de áreas de interés básico	Un resumen del análisis de cada apartado en global.
	Análisis de áreas de interés avanzado	Un desglose del análisis de cada apartado
	Análisis básico trazabilidad cliente	Análisis del comportamiento del cliente en la web
	Análisis avanzado trazabilidad cliente	Análisis pormenorizado del movimiento del cliente en la web
	Análisis cruzado de campañas online	Análisis y comparación del resultado de las campañas
	Análisis avanzado de campañas online	Análisis pormenorizado del resultado de una campaña
	Análisis cruzado de newsletter	Análisis y comparación del resultado de las newsletter
	Análisis avanzado de newsletter	Análisis pormenorizado del resultado de una newsletter
	Análisis cruzado de permission email	Análisis y comparación del resultado de los permission
	Análisis avanzado de permission email	Análisis pormenorizado del resultado de un permission

Tabla. 9: Funcionalidades del Módulo rmN eAnalysis

²¹ En el ultimo punto ampliaremos esta aplicación de TDI

²² Solución interesante que se analizara en el punto 3.4.AnalisisCampañas,newsletter, permission, web...

3.3 Diseño Técnico del Sistema

Para crear una aplicación altamente reutilizable, al desarrollar rmN, se cuidó mucho el modelo de programación.

Este modelo, ahora muy extendido, define al menos tres capas completamente diferenciadas: backend, middleware y frontend.

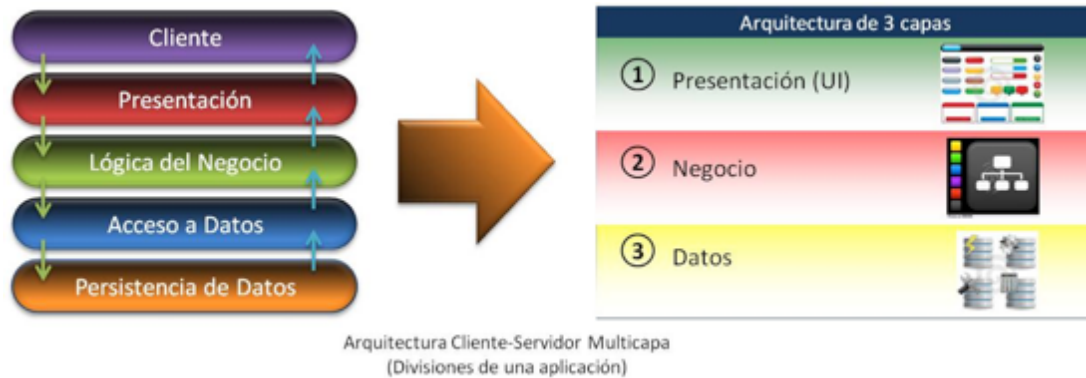


Ilustración 16: Modelo de Programación en Tres Capas

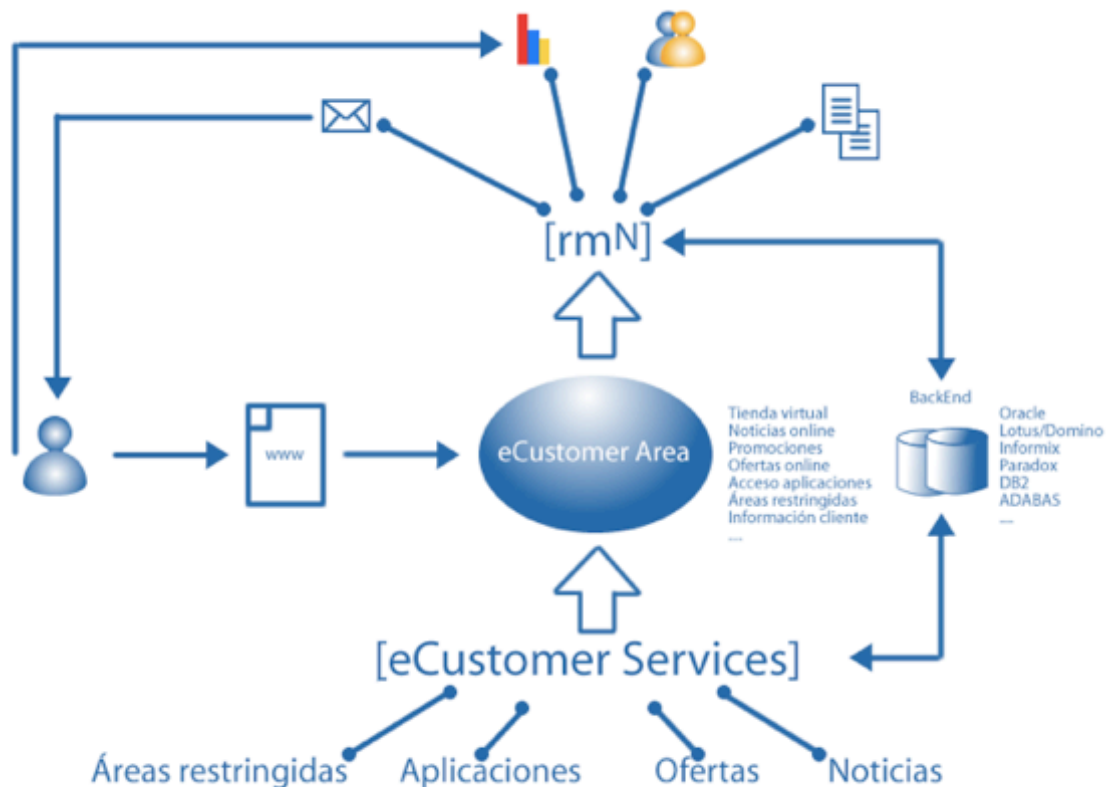


Ilustración 17: Arquitectura Lógica de rmN como parte de eCustomer Services

El **backend** engloba todas las funcionalidades genéricas de bajo nivel que requiera la aplicación. Dentro de este grupo de programas se incluye, sobre todo, el acceso a base de datos; a parte de otras funcionalidades base como el envío de correos, el acceso a

ficheros, la apertura de socket y todas aquéllas que no dependan de un modelo de negocio o producto concreto.

Esta capa debe ser completamente portable, por ello debemos programarla con mucho cuidado, ya que será la base de todas nuestras aplicaciones posteriores. Por tanto, debe estar totalmente desvinculada de cualquier función de negocio y sólo implementar funcionalidades simples, facilitando así el uso de estos recursos (base de datos, envío de correo,, etc.) a capas superiores. Para aplicaciones web, que es el apartado que nos concierne en éste proyecto, el backend no interactúa en ningún caso con el servidor de aplicaciones. Debe estar programado de tal manera que sea lo más estándar posible, especialmente en el aspecto de base de datos, ya que cada cliente puede utilizar una base de datos distinta, e incluso, un mismo cliente, llegar a utilizar distintas bases de datos que deben poderse integrar de manera sencilla a su CRM.

El lugar donde desarrollaremos todas las funciones de negocio es el **middleware**. El middleware se sirve del backend para realizar funciones simples (obtener un dato de la base de datos, enviar un correo a una dirección concreta, guardar una gráfica en un archivo,, etc.), procesa los datos de manera inteligente y los prepara para que el frontend los muestre.

Dentro del middleware, podemos diferenciar una subdivisión: en la parte baja encontramos las funciones específicas del producto, en este caso un CRM. Esta parte baja no debe basarse en las necesidades de negocio de un cliente concreto para que, de éste modo, el producto pueda reutilizarse con futuros clientes sin tener que repetir trabajo. En nuestro caso por ejemplo, el middleware debe cubrir todos los requerimientos de un CRM (estadísticas de clientes, envío de campañas, estratificaciones,, etc.), y no sólo basarse en una empresa de maquinaria.

En la parte alta del middleware encontramos el desarrollo a medida para el cliente, es aquí donde la aplicación comienza a adaptarse a las necesidades de un cliente concreto. Podemos decir que comienza a personalizarse.

El middleware esta en estrecho contacto con el servidor de aplicaciones, ya que es la pieza central del modelo que sirve de cohesión entre el backend y el frontend. Por consiguiente, debe estar abierto a múltiples servidores de aplicaciones para así adaptarse a cualquier situación que pueda darse en el cliente.

En último lugar encontramos el **frontend** o capa de presentación. Este punto une la aplicación con el usuario final, de ahí que deba ser clara y sencilla para cualquier persona, resultar intuitiva y, en la medida de lo posible, atractiva visualmente. Es por ello que se requiere la ayuda de un equipo creativo para desarrollar esta capa, equipo con el cual nos deberemos integrar finalmente. Ha de estar preparada para funcionar en cualquier equipo final, sin importar las características y conexión del mismo, por lo que seguimos la filosofía “thin client” (cliente ligero), es decir, todo el procesado de datos y las tareas que puedan cargar el equipo, se harán en el servidor, el usuario descargará sólo la capa de presentación, que tendrá el menor peso posible.

Resumiendo, el backend es la capa de más bajo nivel donde se implementan las funciones básicas y/o genéricas. Conecta la aplicación con los datos y con otras aplicaciones. Debe ser exportable a cualquier producto y capaz de soportar distintos tipos de base de datos.

El middleware implementa las funciones de negocio. Tiene dos partes diferenciadas, una genérica por producto y otra adaptada al cliente. La primera debe ser altamente exportable. Todo el middleware debe adaptarse perfectamente a cualquier servidor de aplicaciones.

El frontend es la capa visible para el cliente, debe ser muy usable y funcionar correctamente en cualquier equipo.

Adjuntas un par de tablas de los requisitos técnicos del sistema:

Tecnología del lenguaje de programación	10. JAVA: JSP & SERVLETS, TOMCAT SERVER 4.0.6
Base de datos	BDD Integrada: MySQL 4
Comunicación entre sistemas	Comunicación por GPRS LAN
Acceso a la plataforma	Protocolo de actuación: https (Proporcionado por navegadores web convencionales; explorer, netscape, mozilla...).
	Modo de actuación: concurrente multiusuario simultáneo.
Compatibilidad con sistemas externos	Ver tabla.
Capacidad de integración con sistemas preexistentes	Ver tabla
Hosting servidor de aplicaciones	Servidor UNIX 2 GHz, 1024 Mb RAM, 40 Gb HDD.
Requisitos terminales	Conexión a internet/intranet. Aplicación de navegación web pre-instalada.

Tabla. 10: Requisitos Técnicos del Sistema

3.3 Diseño Técnico del Sistema

La plataforma Planet Media rmN 4 es compatible con los siguientes navegadores, sistemas operativos y bases de datos.

<u>Navegadores y servidores de aplicaciones</u>	<u>Sistemas Operativos</u>	<u>Bases de Datos</u>
<ul style="list-style-type: none">• Microsoft Internet Explorer 5 y posteriores• Microsoft Internet Explorer Pocket PC Edition.• Netscape Communicator 4.7x (also on Linux and Unix)• Netscape Navigator 6.2x• BEA WebLogic Server 6.1• IBM WebSphere Server 4.0• Windows NT4• Windows 2000• HP-UX 11 and 11.11(i)• AIX 4.3 and 5.1• Solaris 7 and 8• Tru64 Unix 5.1• Red Hat Linux 7.2	<ul style="list-style-type: none">• Windows 2000• Windows NT4• AIX 4.3, 5.1 and 5.2• HP-UX 11 and 11.11(i)• Solaris 7,8 and 9• Tru64 Unix 5.1• Pocket PC• Windows CE	<ul style="list-style-type: none">• <u>Microsoft SQL Server</u>• <u>IBM DB2</u>• IBM DB2 UDB for OS390• Oracle• Sybase• Informix• MySQL• Adabas• Paradox• Borland

Tabla. 11: Compatibilidad de rmN

3.4 Modelo de Base de Datos

El modelo de datos de rmN es en general bastante sencillo.

La tabla principal es la tabla Clientes, ya que es corazón en torno al que gira una aplicación de CRM.

A su alrededor encontramos la tabla de grupos, que permite unir con Campañas y Newsletter, la tabla de estadísticas de extranet y con los alquileres que ese cliente ha hecho a través de la misma.

Existen una serie de tablas independientes más orientadas a la gestión como usuarios, alertas o noticias.

El punto más creativo del modelo de datos es la forma en la que se guardan las estadísticas de una campaña o newsletter concreta ya que se crea un par de tablas, para newsletter New(# de la Newsletter) LoadStats y New(# de la Newsletter) ClickStats para guardar los datos necesarios para que el módulo de análisis pueda decidir el éxito o no de una Campaña o Newsletter y como esto impacta en el tipo de cliente al que ha sido enviada.

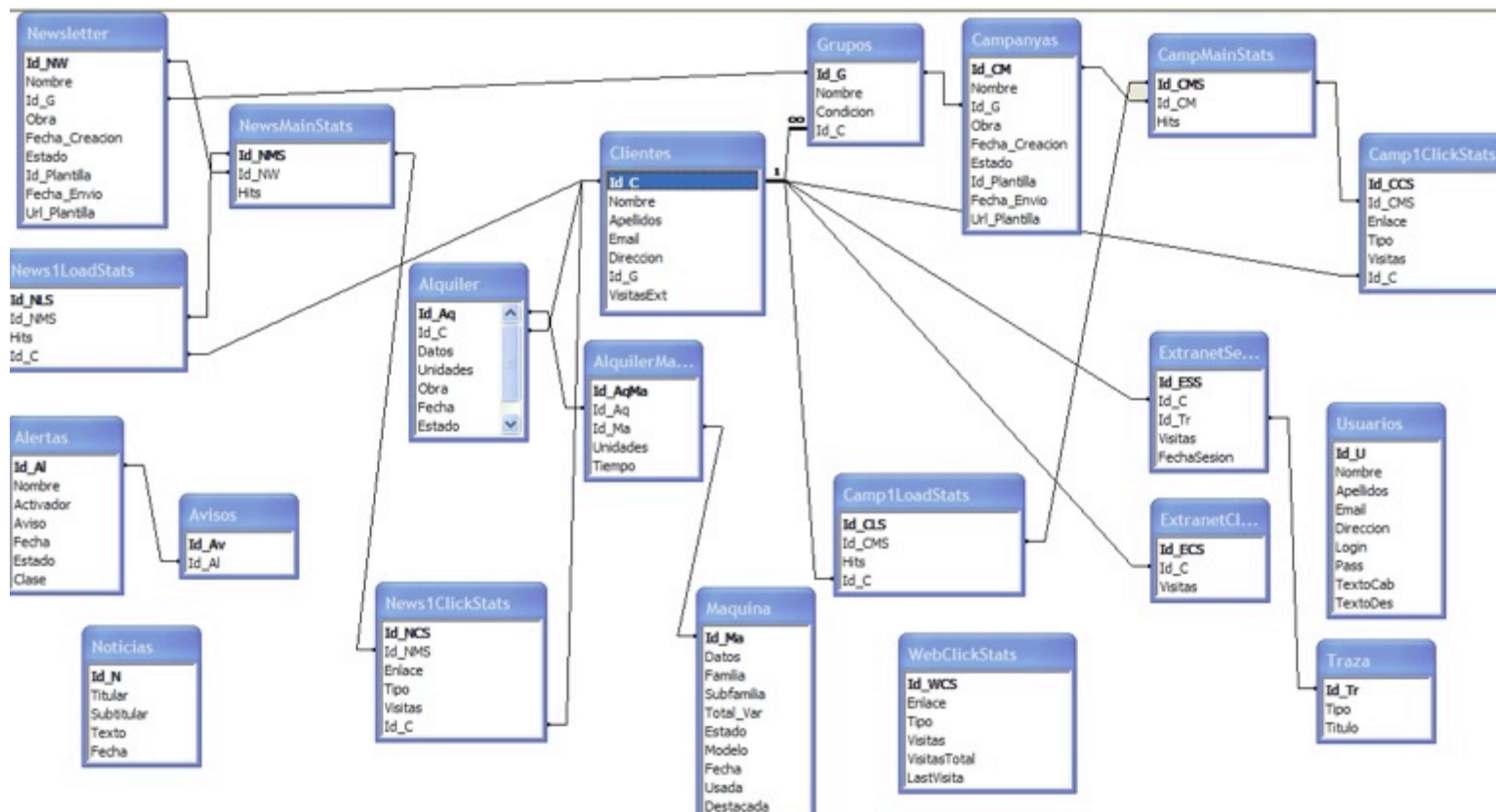


Ilustración 18: Modelo de Datos de rmN

3.5 Desarrollos JAVA destacados

Al ser un producto comercial desarrollado dentro de una empresa para el uso de un cliente no es posible detallar todo el desarrollo realizado durante el proyecto, por esa razón, sólo se hace mención a ciertos desarrollos destacados por su interés técnico objeto de este proyecto y en acuerdo con todas las partes.

3.5.1 Búsqueda iterativa y creación de grupos

Uno de los requisitos básicos impuestos por MERA a hora de gestionar sus clientes era la posibilidad de realizar búsquedas complejas y generar grupos de manera dinámica. Para ello, la solución que se definió fue la creación de un motor de búsquedas iterativo con la posibilidad de formar grupos del resultado de dichas búsquedas.

En rmN 3.0.1 las búsquedas avanzadas se limitaron a las características básicas de los clientes (ciudad, código postal, tipo de empresa, facturación, , etc.) sobre las cuales se podían realizar búsquedas cruzadas. Por ejemplo, se pueden buscar todos los clientes del área de Madrid con un facturación mayor a 100.000 euros anuales.

Así mismo, se pueden utilizar caracteres especiales en las búsquedas, proporcionando así una mayor libertad a la hora de generar las búsquedas y, por consiguiente, a la hora de crear grupos. Por ejemplo, para encontrar el listado de clientes con domicilio en Madrid capital, podemos realizar una búsqueda por código postal, de la siguiente forma: “280%” o “280**”.

Una vez terminada la búsqueda sobre tantos detalles como se desee, se puede crear un grupo y asignarle un nombre de referencia. Una vez creado un grupo, la aplicación no sólo incluirá todos los registros actuales que cumplan los requisitos especificados por el usuario para ese grupo, sino que además, en adelante, incluirá en dicho grupo los nuevos registros que encajen en dicha descripción, manteniendo los grupos constantemente actualizados.

La creación de grupos es básica para la realización de campañas, newsletters, y en general para el aprovechamiento óptimo de la herramienta.

Este motor de búsqueda se ha incluido en todos los módulos en los que la búsqueda ofrecía un valor añadido o facilitaba la gestión de dicho módulo.

3.5.2 Generación de campañas y newsletters

Una funcionalidad importante de rmN es la posibilidad de generación, envío y análisis de campañas y newsletters. En este punto, trataremos de explicar brevemente los aspectos más relevantes de la funcionalidad de ambos módulos (ya que son iguales) sobre

los procesos de creación, envío y contabilización para el análisis. El análisis de campañas propiamente dicho lo veremos en un punto posterior.

La creación de campañas y newsletters (en adelante, todo lo referido a campañas es aplicable de igual modo a newsletters) se apoya en dos pilares básicos, las plantillas y los grupos. De los grupos ya hemos hablado en el punto anterior, reseñando aquí sólo el hecho de que para generar una campaña es condición indispensable asignársela a un grupo. Pasamos pues al estudio de la solución de las plantillas.

Para que se pudieran generar campañas de manera dinámica, pudiéndose insertar cualquier combinación de fotos, textos y adjuntos (limitado al diseño de la plantilla), se pensó e implementó una solución basada en plantillas-tag. Esta solución se compone de una plantilla que contiene el diseño fijado por los creativos “vacío”, es decir, sin elementos fijados, con unos tag en su lugar, de tal forma que pueda ser “rellenado” por el usuario a su antojo, es decir, que el usuario pueda fijar los elementos (fotografías, texto, noticias, ...) que estime en cada campaña.

Para poder crear un interfaz sencillo para el usuario, que no le llevara en ningún caso a un error al generar las campañas, se desarrolló un mecanismo por el cual cuando el usuario inserta una nueva plantilla desde el panel de control de rmN, éste genera un formulario unívoco a esa plantilla, permitiéndole rellenar únicamente aquellos elementos dinámicos de su plantilla.

Una vez subida una plantilla a rmN y creado un grupo, se puede generar una campaña. Para ello se elige el tipo de plantilla entre los diseños existentes, se le asigna un grupo y se “rellena” convenientemente, según estime el usuario. Una vez finalizado el “relleno” de la plantilla se podrá visualizar el resultado provisional de la campaña. Al generarse la campaña, se crea un archivo de campaña que será enviado al grupo seleccionado en el momento que decida el usuario. Este archivo está preparado para servir de vía de comunicación entre el cliente al que le ha llegado la campaña y el módulo de análisis. Esta comunicación es también personalizable durante el proceso de creación de la campaña por el usuario, pudiendo enlazar cada ítem de la campaña a su base de datos (productos, noticias, documentos, ...).

La comunicación se establece a través de dos servlet. Uno de ellos gestiona las aperturas de la campaña, que se resolvió haciendo que en cada apertura del correo, éste pidiera un elemento “dot” a través del servlet al servidor para almacenar su apertura. El otro servlet gestiona los accesos a la información que el usuario ha incluido en la campaña. Toda esta información se almacena en dos de tablas que se generan dinámicamente al enviar la campaña y que sirven, una para guardar las aperturas de una campaña en concreto y la otra para guardar los elementos de una determinada campaña que ha abierto un cliente en concreto.

Estas dos tablas por cada campaña serán utilizadas por el módulo de análisis para determinar la efectividad de una campaña o newsletter.

3.5.3 Campañas y newsletters

Para el análisis de campañas y newsletters son necesarias dos tablas que se generan en el momento de enviar la campaña o newsletter. Una de las tablas nos servirá para guardar todas las aperturas por cliente de dicha campaña/newsletter. Dicha acción se detecta mediante la carga desde el servidor a través de un servlet de una imagen cada vez que es abierta la campaña/newsletter.

La segunda tabla es usada para almacenar las ampliaciones de información, es decir, las visitas a la página web a través de la campaña o newsletter enviada. El proceso de captura por cliente se realiza mediante un servlet embebido en el hipervínculo de cada link.

Por tanto, es aquí donde podemos ver que el modelo de datos de rmN es dinámico, puesto que requiere de la creación de dos tablas para el análisis de cada campaña o newsletter que se envíe. Posteriormente a la captura de los datos, y ya desde rmN, se podrán analizar los resultados del impacto de la campaña. Para dicho análisis, se han utilizado fórmulas basadas en el tratamiento digital de la información (TDI), las cuales explicaremos en un punto dedicado exclusivamente a ellas.

A la hora de presentar los datos del análisis, se pueden visualizar dentro de rmN en modo tabla o en modo gráfico. Para ello, se integraron componentes programados en flash con componentes propios de JAVA.

El análisis de estas campañas y newsletters, contribuye al análisis final de clientes y productos.

3.5.4 Web y extranet

Para el análisis de la web, todos y cada uno de los links relevantes de la misma están conectados a través de un servlet que almacena las entradas de dichos links. Después, la información es presentada desde el módulo de análisis de rmN en una tabla en forma de árbol integrada con flash en la que se presentan todos los productos que se ofrecen a través de la web ordenados por “familia” y “subfamilia” incluyendo la información del número de cargas por producto. Otra información interesante de la web es el listado de links más visitados, donde se incluyen noticias y el resto de posibilidades de la página.

También merece la pena destacar el asistente de maquinaria que ayuda al navegante a elegir el modelo que mejor se adapta a sus necesidades. Desde el módulo de análisis de rmN, podemos consultar los resultados más solicitados al asistente, ya que un servlet guarda la información que le provee el flash del asistente en una base de datos preparada para ello.

En la extranet, la información que se guarda es la misma que en la web, con el aliciente añadido de poder discriminar qué cliente es el que realiza la visita. Se puede hacer un seguimiento completo del viaje de un cliente a través de la extranet en una gráfica que provee rmN también integrada con flash.

La información recogida en la extranet no sólo es útil a nivel de producto sino, como ya hemos comentado, será un punto a tener en cuenta a la hora de clasificar a los clientes.

La implementación del módulo de análisis de web y extranet de rmN es muy sencilla y nos puede ayudar a mejorar cuantitativamente el aprovechamiento de este recurso que es la red de redes.

3.5.5 Productos y clientes

Sin duda la parte del análisis más importante, y por ello más interesante, es la referente a los clientes y los productos.

Sobre los productos, gracias a toda la contabilización que se realiza entre la web, la extranet y las campañas y newsletters, se elabora un ranking de aquellos que tienen mayor acogida por los clientes.

Tenemos que añadir a estos datos los resultados del asistente que programamos para la extranet vinculada con el CRM. Todos los datos sobre los productos están guardados en una tabla preparada par ello y las entradas vienen todas a través del servlet que contabiliza las consultas al producto.

De todos los análisis, el de los clientes es el análisis estrella.

Para analizar el interés de un cliente se tiene en cuenta el comportamiento del mismo a través de las acciones que efectuamos sobre él, que son las campañas, las newsletters y la disponibilidad de una extranet. Dentro de cada una de ellas analizamos tanto la visita de un cliente como el comportamiento dentro de la visita. Sobre todos los datos recogidos se aplica una función lineal de clasificación basado en TDI cuyos coeficientes han sido determinados mediante un proceso de entrenamiento. Los parámetros que son medidos para determinar el interés de un cliente son:

- N° de campañas recibidas.
- N° de aperturas de las mismas.
- N° de productos visitados a través de la campaña.
- N° de newsletters recibidas.
- N° de aperturas de las mismas.
- N° de productos visitados a través de la newsletter.
- Frecuencia de entrada en la extranet.
- Tiempo en la extranet.
- Productos visitados.

El proceso de obtención de la fórmula aplicada y sus coeficientes se explicarán con detalle en un apartado posterior. Todos los resultados sobre la determinación del impacto de las campañas, newsletters, extranet, etc. se calculan en base a sendas formulaciones basadas en TDI y obtenidas por medio de entrenamiento lineal.

NOTA: Para que percibir el impacto de una campaña o el interés de un cliente fuera más rápido y sencillo, se ideó un sistema de estrellas. Este sistema no es más que la presentación para el usuario en cada ítem analizado de una estrella con más o menos opacidad según la puntuación (también reflejada en la estrella) del 1 al 10.

3.5.6 Alertas web, email y sms

Existen 2 tipos básicos de alertas, las referentes a las acciones propias de la aplicación y que son configuradas a través de la herramienta de administración de rmN (alertas automáticas) y las personales.

Las alertas personales se presentan poco interesantes desde el punto de vista que analizamos en este apartado. El usuario solo tiene que poner la fecha y el texto y la herramienta se encarga de avisarle. Pero abren un amplio abanico de posibilidades para la mejora de la herramienta en futuras versiones.

Para avisar al usuario de cualquier alerta, en el proceso de login se testea la tabla de alertas y se presenta una alarma en la página principal de rmN para informarle de que tiene nuevas alertas. Esta alarma se repetirá en cada login hasta que sea apuntada como vista.

Las alertas automáticas tienen que ser configuradas y activadas por el administrador de rmN. Cada vez que se produce una alarma de las predeterminadas se hace una llamada al servlet que se encarga de incluirla en la base de datos. Dentro de este tipo de alertas se pueden distinguir dos clases: las referentes a acciones básicas (entrada de un nuevo alquiler, entrada de un nuevo usuario, envío de una newsletter, ...) y las referentes al análisis de rmN (tal campaña ha alcanzado “n” visitas, tal cliente ha pasado de “bueno” a “muy bueno”, tal producto a recibido “n” visitas, ...).

Estas alertas automáticas son limitadas y vienen como input en el briefing.

3.5.7 Administración de rmN

La herramienta rmN es configurable mediante una aplicación con la misma misma apariencia a la que se accede a través de una entrada con un login y password distintos al de rmN, y que deberían ser conocidos únicamente por el administrador de la herramienta.

Desde esta herramienta se pueden crear los usuarios de acceso a rmN, subir las plantillas tanto de envío de correos como de campañas y newsletters y pueden ser configuradas las alarmas que se desean tener en cuenta durante el uso de la herramienta, asignándoles nombre y texto que luego se podrá visualizar desde rmN.

Es en el momento de subir las plantillas cuando la aplicación crea el formulario de campaña o de newsletter que se ajusta unívocamente a la plantilla que se sube. Durante este proceso es chequeada la integridad de la plantilla y guardada en el directorio correspondiente, creando una entrada en la base de datos para ella.

3.5.8 Permission e-mail

Gracias al Permission e-mail el usuario será capaz de importar BBDD de cualquier tipo y forma desde un sencillo grupo de formularios.

Desde los formularios se van trayendo las BBDD, las tablas y los datos siempre que se completen los datos del primer paso, url de conexión, login y password y el tipo de BBDD para importar el driver de conexión necesario.

Capítulo 4

Tratamiento Digital de la Información

4.1 Ensayos preliminares

Antes de realizar toda la programación sobre un decisor en concreto para su implementación en la plataforma CRM, durante el proyecto analizamos varios para ver cual era el que mejor resultado nos podía dar, partiendo siempre del supuesto del uso de un método lineal que fuera fácilmente implementable y adaptable.

Si bien el estudio de la Universidad de Valencia [13] nos sugería que para realizar una categorización, en este caso binaria, es decir, decidir si un individuo iba a ser o no un buen cliente, se comportaría mejor una red neuronal, no quitamos del estudio el regresor para que nuestro estudio fuera completo.

En esta parte del proyecto analizó el regresor lineal, el Perceptrón y el Adaline para decidir el mejor método que luego utilizaríamos en nuestro desarrollo para encontrar los pesos de nuestro decisor.

4.1.1 Definición del Problema

Inicialmente no podíamos asegurar que los datos que obtendríamos fueran totalmente separables con una frontera lineal, por tanto, el caso de la implementación de métodos lineales podemos prever que el modelo que encontrará los pesos más cercanos a los óptimos será el Adaline. No será el Perceptrón debido a que como los datos no son linealmente separables no convergerá y tendremos que forzar su detención; no obteniendo así los pesos óptimos. No será el Regresor lineal + umbral porque este no ponderará correctamente las muestras ya que tomará como límite inferior y superior infinito y no -1 y 1 como debería corresponder en este caso.

El algoritmo Adaline obtendrá mejores resultados tanto si la muestra es linealmente separable como si no, ya que no se basa en minimizar el número de errores existentes en los datos de entrenamiento. Lo que se busca minimizar con Adaline es la desviación de la red para todos los patrones de entrada, eligiendo una medida del error global. Normalmente se utiliza el error cuadrático medio. Si podemos predecir que su resultado será parecido a realizar una minimización de errores. Otro motivo es que pondera los datos mejor que los dos otros métodos, proponiendo para esto una función de activación blanda que en este caso se trata de la tangente hiperbólica.

Para comparar unos métodos de búsqueda con otros miraremos la relación de la Probabilidad de Detección con la Probabilidad de Falso Negativo (Curva OC²³) y la probabilidad de error que genera cada uno de los métodos.

La curva OC nos indica cada uno de los puntos de trabajo en los que el método se puede encontrar. Por ejemplo, si queremos minimizar la probabilidad de falso negativo hasta un determinado valor, con esta curva podemos saber que probabilidad de detección se obtendría, y viceversa para la maximización de la probabilidad de detección.

Para tomar la decisión de qué método es mejor con respecto a esta curva tendremos que observar los valores de un punto de trabajo para las curvas que se comparen (para todas el mismo punto de trabajo) y ver qué valores se obtienen . Asimismo, respecto a la probabilidad de error nos basaremos en el mínimo de la función de error. Teniendo en cuenta solo la probabilidad de error el método cuyo mínimo sea inferior será aquel que obtenga los mejores pesos.

4.1.2 Métodos Lineales Analizados.

En la **Figura 19** se puede ver la máquina con la que vamos a implementar los modelos lineales. Llamamos a esta estructura decisor lineal porque implementa fronteras lineales. Los decisores lineales que hemos implementado tienen sesgo (w_0), esto permite que puedan no pasar siempre por el origen de los ejes.

El problema que resuelve cada uno de los algoritmos será del estilo:

²³ OC del inglés Operational Characteristic, Características Operacionales, se refiere a graficas que ayudan a determinar la calidad de un producto, de la señal de un emisor, etc. Basándose en la probabilidad de acierto y de error, como la aparición de una lata deformada y que esta no se retirada.

$$w_0 + w_1 \times x_1 + w_2 \times x_2 + \dots + w_n \times x_n = 0$$

Donde w_n representa los pesos, los coeficientes de la función y x_n las variables de la misma.

Esta expresión no es una ecuación lineal, aunque si implementa una frontera de decisión lineal.

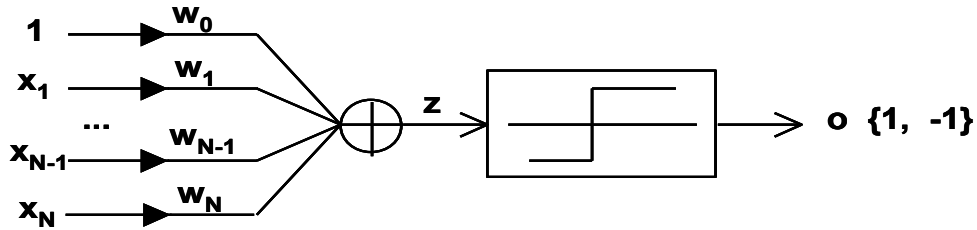


Ilustración 19: Máquina de Modelos Lineales

A la hora de construir la máquina de la **Figura 19** hemos de tener en cuenta que para construir la frontera es necesario como primer paso hacer un diseño general para posteriormente estudiar con más atención las muestras más cercanas a la frontera, ya que estas son realmente las que van a tener más influencia al definir la frontera.

4.1.2.1 Regresor lineal + umbral

La regla utilizada para encontrar los pesos en este caso será la pseudoinversa de Moore-Penrose:

$$W_e = (X_e^k X_e) X_e^k$$

Donde:

- $W_e^{(k)}$: Vector de pesos actual, k
- X_e^k : Vector de variables en el momento k

Para obtener esta regla de obtención de pesos hay que suponer que la frontera va a ser una combinación lineal de las entradas más una variable aleatoria, llamada residuo; la función de densidad de este residuo será una gaussiana de media cero y varianza contante, con lo que cumplirá el criterio de homocedasticidad. Para obtener la expresión hay que minimizar el sumatorio de todos los cuadrados de los residuos. Al hacer esta minimización estamos aproximando la frontera a una combinación lineal de las entradas; para esto hay que tener en cuenta que los residuos son ortogonales a las entradas.

Este algoritmo obtiene buenos resultados cuando nos encontramos ante una situación de entradas deterministas.

4.1.2.2 Decisor basado en la regla del Perceptrón

La regla utilizada para encontrar los pesos óptimos sería:

$$W_e^{(k+1)} = W_e^{(k)} + \frac{\alpha}{2} (d^k - o^k) \times X_e^k, \quad 0 < \alpha < 1$$

Donde:

- $W_e^{(k+1)}$: Vector de pesos siguiente a k
- $W_e^{(k)}$: Vector de pesos actual, k
- α : Tasa de aprendizaje, constante entre 0 y 1
- d^k : Salida esperada para la posición k
- o^k : Salida obtenida en la posición k
- X_e^k : Vector de variables en el momento k

Esta regla es la regla del Perceptrón en el caso “on line”; es decir, en caso de que vayamos calculando de forma secuencial los pesos.

Este algoritmo utiliza lo que se denomina procedimiento por refuerzo hebbiano, que consiste en que solo se modifican los pesos si la entrada generada no se corresponde con la salida esperada. Este método tiene varios inconvenientes. El primero es que si el conjunto de datos no es linealmente separable el algoritmo no llega nunca a converger porque se considera que una frontera lineal no es capaz de separar estos datos. El segundo de ellos es que si el problema es linealmente separable el algoritmo puede presentar dos inconvenientes. El primero de ellos es que llega a converger en un número de pasos finito, pero aunque este número sea finito puede ser muy elevado para poder esperar a que el algoritmo converja. El otro inconveniente es que en este tipo de casos normalmente el algoritmo en vez de generalizar memoriza. Esto es debido a que este algoritmo minimiza el número de errores sobre el conjunto de entrenamiento, esto puede llevar de forma gradual a situaciones de sobreajuste.

4.1.2.3 Adaline con activación blanda

El resultado de este algoritmo es cualitativamente parecido a minimizar el número de errores. La ventaja de este algoritmo es que si la tasa de aprendizaje se maneja correctamente el algoritmo converge y el número de pasos se puede controlar con la tasa de aprendizaje. Otra ventaja de este algoritmo es que generaliza mejor que el Perceptrón, a cambio de lo cual el entrenamiento de este algoritmo no se dirige a minimizar el número de errores.

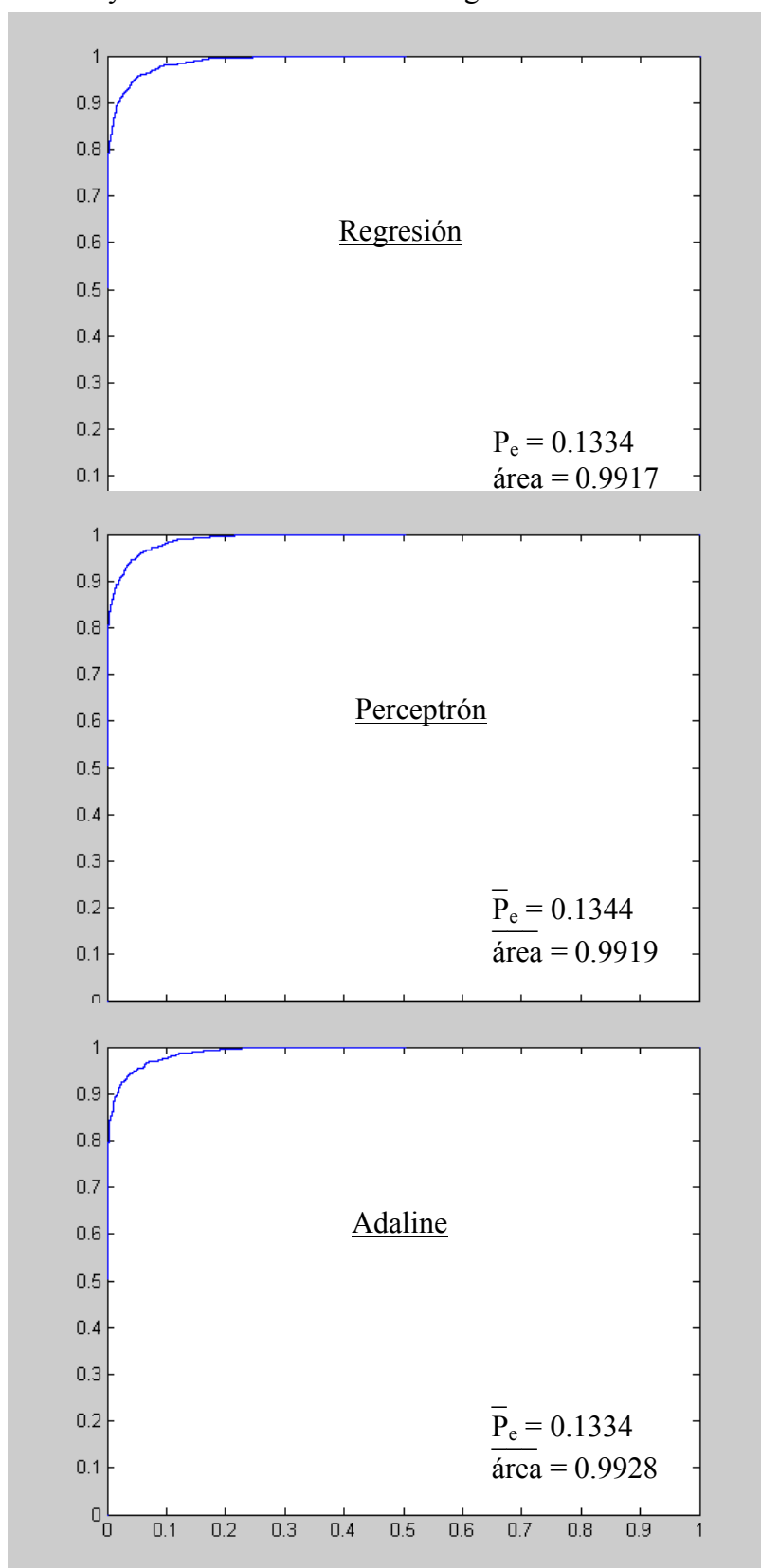
Para realizar el algoritmo con activación blanda pasamos la función z por la tangente hiperbólica. Con esto conseguimos una aproximación de la solución. Esta tangente hiperbólica se utiliza porque la derivada de esta se puede poner a partir de la salida de la no linealidad. Otra razón es que si suponemos que todas las densidades de probabilidad de las hipótesis son gaussianas con diferentes medias y la misma matriz de covarianzas, este algoritmo tiene la suficiente capacidad expresiva para resolver el problema. La última razón es que se puede entrenar al mismo tiempos todos los componentes de una estructura multicapa.

El algoritmo de entrenamiento utilizado para este caso será el siguiente, con una tasa de aprendizaje de 0.001 para que el algoritmo se pueda ir acercando al mínimo progresivamente si n pasárselo:

$$\varpi^{(k+1)} = \varpi^{(k)} - (\alpha \times (d^k - o^k) x (1 - o^2)) \times x^k, \quad 0 < \alpha < 1$$

Las comparativas están basadas siempre en las curvas OC y en la probabilidad de error media encontrada (P_e). Dado que las curvas son difíciles de comparar debido a sus irregularidades, hemos buscado una medida analítica en la que basarnos. Dicha medida es una aproximación de la integral de Riemann: para cada uno de los valores de P_{FN} (Probabilidad de Falso Negativo), se calcula el área comprendida entre el valor de P_{FN} actual y el anterior. Dicho de otra manera, se calcula un área que acotará por encima al área de la curva.

Las curvas y valores obtenidos son los siguientes:



Vemos que:

$$P_{e|perceptrón} > P_{e|regresión} = P_{e|adaline}$$

$$Area_{regresión} < Area_{perceptrón} < Area_{adaline}$$

Por tanto, podemos decir que el Adaline ofrece mejores prestaciones, seguido por los otros dos métodos, de los cuales el cálculo por regresión minimiza la P_e y la búsqueda mediante Perceptrón maximiza la P_D probabilidad de detección ya que minimiza la de falso negativo.

4.2 Aplicación del Tratamiento Digital de la Información al mundo del Marketing

Una vez llegada a la conclusión de que los mecanismos de decisión lineal más precisos son los basados en redes neuronales como vimos en el estado del arte después del estudio de [13] y unido a las pruebas realizadas sobre el Perceptrón y el Adaline del punto anterior, decidimos basar nuestra solución en este último.

Lo que pretendíamos dar a MERA con este módulo es una forma sencilla, basada en el comportamiento de sus clientes y de una forma automática, de un mecanismo para decidir sobre que individuos invertir una mayor cantidad de esfuerzo comercial ya que, según nuestro informe, iban a tener más probabilidades de convertirse en clientes.

Para ello, lo primero que identificamos fueron las variables con las que crear la formula de pesos.

Debían ser suficientemente específicos, fáciles de medir y que aportaran información valiosa.

Puesto que quería que todo fuera automático, nos basamos sólo en datos que pudiéramos capturar directamente con rmN.

Las variables (x_n , en la formulación teórica) elegidas fueron:

- #Accesos a campañas online. AC.
- #Clicks en vínculos de las campañas online. CC.
- #Accesos a newsletter. AN
- #Clicks en vínculos de newsletter. CN.
- #Visitas a la Extranet. VE.
- #Hits en la navegación por la Extranet. CE

Con estas variables, la formula resultante es:

$$w_0 = w_1 \times (AC + AN) + w_2 \times (CC + CN) + w_3 \times VE + w_4 \times CE$$

$$Y_{calc} = \begin{cases} -1, & w_0 < 0 \\ 1, & w_0 \geq 0 \end{cases}$$

Donde:

$$Inversión = \begin{cases} si, & Y_{calc} = 1 \\ no, & Y_{calc} = -1 \end{cases}$$

Capítulo 4: Tratamiento Digital de la Información

Para la captura de datos, tuvimos que implementar un método especial en todas las campañas, newsletter y cada vínculo de la extranet de Mera para que se fuera alimentando nuestra base de datos.

Luego, cruzamos esa información con aquellos individuos que se habían convertido en clientes en un par de meses, uno para el entrenamiento y otro para comprobarlo, y así establecer los pesos que sirvieran para clasificar al siguiente individuo como potencial cliente o como individuo medio.

La cantidad de datos recogida fue bastante corta y tuvimos que extrapolar algunos datos para poner obtener los primeros pesos que luego se irían refinando con retroalimentación con el paso del tiempo.

Este es un ejemplo de parte de los datos recogido. Al ser datos de clientes reales, está sujeto a la LOPD²⁴ por lo que sólo se pueden mostrar los datos:

	W0	(AC + AN)	(CC + CN)	VE	CE	Y
Cliente 1	-0,52	0,13	0,78	0,30	0,41	-1
Cliente 2	-2,66	0,00	0,60	0,10	0,15	-1
Cliente 3	0,01	0,25	0,75	0,43	0,49	1
Cliente 4	-2,59	0,25	0,70	0,03	0,04	-1
Cliente 5	-1,36	0,25	0,55	0,47	0,33	-1
Cliente 6	-2,39	0,50	0,33	0,47	0,19	-1
Cliente 7	-1,48	0,50	0,43	0,27	0,35	-1
Cliente 8	-0,34	0,00	0,70	0,47	0,51	-1
Cliente 9	-3,14	0,33	0,50	0,03	0,03	-1
Cliente 10	-1,52	0,25	0,70	0,27	0,23	-1
Cliente 11	-3,75	0,00	0,50	0,00	0,00	-1
Cliente 12	-1,63	0,17	0,43	0,43	0,37	-1
Cliente 13	-0,06	0,50	0,68	0,27	0,48	-1
Cliente 14	-1,34	0,25	0,55	0,23	0,37	-1

Tabla. 12: Datos Recogidos para el entrenamiento del Adaline

²⁴ Ley Orgánica de Protección de Datos de Carácter Personal de España // http://es.wikipedia.org/wiki/Ley_Org%C3%A1nica_de_Protecci%C3%B3n_de_Datos_de_Car%C3%A1cter_Personal_de_Espa%C3%B1a

4.2 Aplicación del Tratamiento Digital de la Información al mundo del Marketing

Con un par de conjuntos de datos como estos entrenamos nuestro Adaline utilizando una programación en Matlab que se puede consultar en el Anexo 3 y obtuvimos los siguientes pesos:

- $w_1 = -0,7$
- $w_2 = 2,1$
- $w_3 = 0,9$
- $w_4 = 3,15$

La aplicación del desarrollo en Matlab incluido en el Anexo 3 sobre las muestras de datos proporcionadas por MERA arrojaron el resultado de la siguiente figura, donde el eje x contiene el número de individuo analizado y sobre el eje y se representan en rojo el resultado real proporcionado por MERA, es decir, si resultó ser un cliente, en azul el w_0 y el verde la decisión:

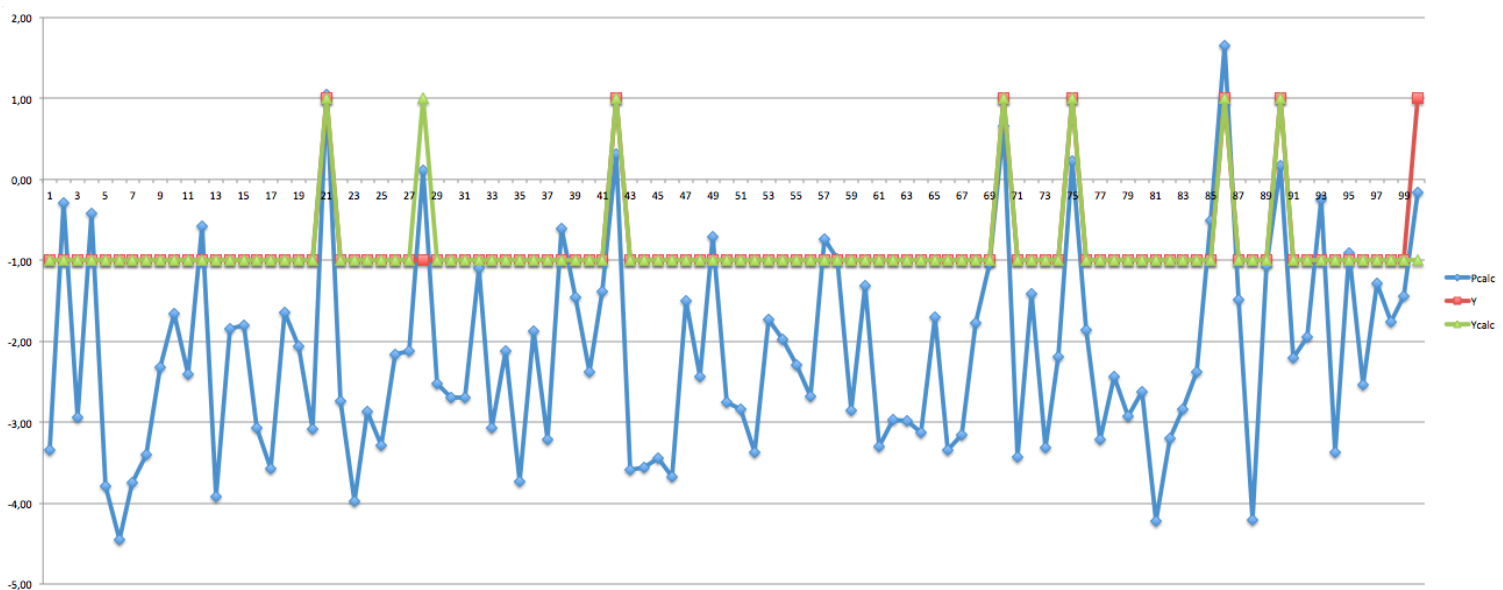


Ilustración 20: Resultados de la Aplicación de Adaline a los Datos de MERA

Se puede ver en azul los valores de w_0 , en rojo la Y , es decir, la opinión de MERA sobre si había merecido la pena o no invertir esfuerzo comercial sobre ese individuo, o lo que es lo mismo, si se había convertido en cliente y el verde está la decisión, llamada Y_{calc} , que es el resultado de aplicar la fórmula con los w_n resultantes del proceso a los datos de test.

Desde el punto de vista de negocio, los fallos realmente catastróficos son aquellos donde la máquina entrenada desechara una oportunidad que luego se hubiera convertido en cliente. Un falso negativo.

En cambio, un falso positivo sólo añade un pequeño sobre coste que, comparado con el anterior estado donde realizaban esfuerzos aleatorios a poblaciones de individuos mayores, es algo mucho más que asumible.

Por ello, cambiamos algo nuestro diseño en lo concerniente a la frontera de decisión, y en vez de fijarla en 0 la fijamos en -1.

Si volvemos a la grafica anterior, se puede observar un solo falso negativo, justo el último, y se demuestra fácilmente que con el nuevo umbral, sí hubiera sido un individuo que se hubiera seguido.

Es fácil entender el beneficio principal que obtuvo MERA de este estudio aplicado al CRM. En una muestra de 100 individuos, MERA ponía un foco medio, es decir, un presupuesto fijo igual para los 100, dado que no tenía una forma analítica de segmentar sus clientes.

De esa metodología de trabajo, gracias al módulo de análisis que desarrollamos dentro del proyecto fin de carrera en el producto rmN, se pasó una inversión más selectiva y mucho más eficiente ya que le mostramos un espectro de menos de 20 individuos con mucho mayor potencial. Aplicando una regla de 3 simple, se observa que pueden dedicar cinco veces más presupuesto a un grupo selecto de individuos o atacar a cinco veces más individuos con alta probabilidad de convertirse en clientes con el mismo presupuesto..

Capítulo 5

Conclusiones y Líneas de Trabajo Futuras

5.1 Conclusiones

La primera conclusión que se puede obtener del proyecto es su éxito, ya que este producto, rmN, fue implantado satisfactoriamente en al menos dos clientes mientras estuve en Planet Media y aun sigue formando parte del offering de esta compañía en el área de CRM.

Sobre la tecnología utilizada, Java, como vimos en el estado del arte, todo el trabajo realizado en su momento sigue teniendo vigor tanto como idea de proyecto, así como la tecnología JAVA, la cual, se podría actualizar y gracias a su portabilidad reutilizar casi al completo para cualquier migración.

Gracias a la facilidad de programación conseguimos un producto modular que cumplía con todas las características. Programado 100% web es fácilmente configurable y compatible con todos los navegadores.

La arquitectura en 3 capas demostró su robustez al no tener ningún fallo durante el tiempo que estuve dando soporte.

La investigación sobre los diferentes decisores nos llevó a la conclusión de que Adaline, para este caso en concreto, daba mejores resultados que el Perceptrón o la regresión lineal.

Al ser un problema los falsos negativos, pudimos adaptar nuestro desarrollo para que no se perdieran individuos con alto potencial, lo que demuestra que todo estudio teórico necesita un ajuste dependiendo de la realidad a la que se enfrenta.

Los conceptos de tratamiento digital de la información que pude aplicar, los he seguido aplicando a otros sectores, como ahora mismo que me encuentro en mitad de una consultoría para mejorar el sistema de peticiones del Santander y donde estoy proponiendo un sistema de asignación que se base en el aprendizaje y en base a él ayude a la categorización de problemas en base a una serie de variables.

Sin duda fue un proyecto que nos ayudó a poner en práctica muchas cosas vistas durante la carrera, tanto a nivel de programación, de tratamiento digital de la información así como de gestión de proyectos ya que al trabajar para un cliente final, se tienen que tener muy en cuenta las planificaciones, seguimiento y gestión.

Creo que el espíritu imbuido en mí durante mis estudios universitarios me sirvió para llevar a cabo las investigaciones pertinentes para asegurar que en el proyecto estamos utilizando las mejores opciones disponibles y ejecutarlo de forma correcta, así como dar ese valor en el módulo de análisis y decisión.

5.2 Futuras líneas de desarrollo

1. Nueva tecnología. Lógicamente, con todo lo que ha evolucionado el mundo de la programación, se podrían obtener muchos beneficios actualizando tecnológicamente la plataforma. Mejor gestión de la base de datos, gráficos con más, utilizar información desestructurada, redes sociales...
2. Política de usuarios. (Perfiles). Actualmente sólo hay un tipo de usuario, administrador, de acceso al sistema. Podrían crearse otros como Director de Marketing, Agente, Junior, etc.
3. Modularización de la interfaz. (El frontend en módulos, laterales, central...). La parte web de acceso al usuario está creada como un solo jsp. Podría ser una forma de simplificar, reusar y mejorar el aspecto si se separan.
4. Planificador rmN. Desarrollar un módulo que permitiera hacer seguimiento de la actividad de marketing sobre los clientes, con un planificador de tareas y seguimientos de las mismas sobre distintos perfiles....
5. Creación y configuración de alarmas automáticas. Desarrollar un módulo que utilizara la información del módulo de análisis para crear alarmas. Por ejemplo, cuando un individuo pasa de ser un individuo medio a un potencial cliente. O campañas que han tenido más éxito que otras.
6. Mas y mejores variables para clasificación. Se podrían añadir nuevas variables a la fórmula de cálculo del análisis de clientes, como encuestas, reenvíos de la campaña...
7. Otros métodos de clasificación. Aquí hemos utilizado Adaline, pero se podrían explorar otros métodos, quizá no lineales, que con otras variables sean más útiles.
8. rmN as a service. Por último, estando de moda el Cloud y los modelos de pago por uso de aplicaciones, se podría plantear una solución de CRM que sea contratada y consumida en dicha modalidad...como software como servicio.

Capítulo 6

Referencias y Otras fuentes

6.1 Referencias

- [1] *Andrés Vegas Aneiros. CRM en Internet.* (s.f.). En capítulo 1. Recuperado el 30 de 08 de 2013, de http://www.marketingycomercio.com/numero15/00jul6_crm.htm
- [2] *Reinales Lara, Pedro J. y Ponsoa Casado, Jose Manuel, Marketing Relacional: Un nuevo enfoque para la seducción y fidelización del cliente, Editorial Times-Prentice Hall, Segunda Edición, México 2002, p. 128*
- [3] *Universidad Politécnica Salesiana.* En capítulo 1. Recuperado el 30 de 08 de 2013, de <http://dspace.ups.edu.ec/bitstream/123456789/585/3/CAPITULO%20II.pdf>
- [4] *Profesor Enrique Dans, Crm - Custom Relationship Management: Versión original de 10 de enero de 2002. Última revisión, 28 de septiembre de 2006. Editado por el Departamento de Publicaciones del Instituto de Empresa.*
- [5] *Estado del Arte_Java: Sun Microsystems.* Recuperado el 05 de 09 de 2013, de:
www.sun.com
<http://www.javasoft.com>
- [6] *Estado del Arte_Java: Java en castellano.* Recuperado el 05 de 09 de 2013, de:
www.programacion.com/java/
- [7] *Estado del Arte_Java: Revista electrónica sobre Java.* Recuperado el 05 de 09 de 2013, de:

<http://www.javaworld.com/JavaWorld>

- [8] J. Manger: *Java: More than just a programming language*. *Internet Business Magazine* (April 1997), p. 66-70.
- [9] G. Alwang, J. Clyman, R.V. Dragan, L. Seltzer: *Java Guide*. *PC Magazine*, vol.17 no. 7 (April 1998), p. 101-209.
- [10] P.M. Cuenca Jiménez: *Programación en JAVA para Internet*. *Anaya Multimedia* (col. *Vía@Internet*), Madrid, 1996. xxiv+528p.
- [11] R.V. Dragan, L. Seltzer: *Java: A Field Guide for Users*. *PC Magazine*, vol.16 no. 10 (May 1997), p. 100-115.
- [12] J.M. Framiñan Torres: *Java*. *Anaya Multimedia* (col. *Al Día en una Hora*), Madrid, 1997. 128p.
- [13] Alfonso Pitarque, Juan Francisco Roy y Juan Carlos Ruiz: *Redes neurales vs modelos estadísticos: Simulaciones sobre tareas de predicción y clasificación*. 1998. *Universidad de Valencia*.

6.2 OTRAS FUENTES

6.2.1 Artículos:

- "Permission Marketing", ABC, 22 de Julio de 2001.
- "Un poco de CRM en la ensalada, por favor", Cinco Días, 9 de junio de 2001.
- "Sobre modas y realidades: CRM o el nuevo marketing digital" (45 Kb.) Información Comercial Española (ICE), número especial sobre "Nueva Economía y Empresa", n. 791, abril-mayo 2001, pp. 55-62.
- "Filtros colaborativos vs. sistemas basados en reglas", ABC, 11 de febrero de 2001.
- "Cookies: Identificando al usuario", ABC, 19 de noviembre de 2000.
- "¿Cuánto vale el loro?", Cinco Días, 24 de junio de 2000.
- "CRM, Customer Relationship Management", ABC, 7 de mayo de 2000.
- R. Gilbertson: NC's the Fourth Wave — The Network computer is switching the client-server standard to server-client. Tech Web News (June 16, 1997).
- INN Touch (boletín informativo periódico de Innovative Interfaces Inc.). Números de 1997 y 1998 (muchos de ellos dedicados a Java o al sistema de gestión de Bibliotecas INNOPAC Millennium). Pueden consultarse sus versiones electrónicas en la dirección <http://www.iii.com>
- P. Jones: Java and Libraries. Digital and Otherwise. D-Lib Magazine (March 1997). [Se trata de un artículo de una revista electrónica, que puede consultarse p. ej. en <http://mirrored.ukoln.ac.uk>; su versión impresa ocupa unas 5p.]

6.2.2 Publicaciones

Permission Marketing, Seth Godin, 1999.

Escrito por el ex-VP de Marketing de Yahoo! define el concepto de permission marketing y ahonda en cuáles son los niveles de permiso que una empresa puede tener con sus clientes, cómo incrementarlos y cómo manejarlos adecuadamente. Hay una versión en castellano editada por Granica.

The One-to-One Future, Don Peppers and Martha Rogers, 1996.

El pionero del género, que sigue plenamente vigente a pesar de haber pasado treinta "años Internet". Indispensable para cualquier interesado no sólo en el CRM, sino simplemente en el marketing como tal.

J. Manger: Java: **More than just a programming language**. Internet Business Magazine (April 1997), p. 66-70.

G. Alwang, J. Clyman, R.V. Dragan, L. Seltzer: **Java Guide**. PC Magazine, vol.17 no. 7 (April 1998), p. 101-209.

P.M. Cuenca Jiménez: **Programación en JAVA para Internet**. Anaya Multimedia (col. Vía@Internet), Madrid, 1996. xxiv+528p.

R.V. Dragan, L. Seltzer: Java: **A Field Guide for Users**. PC Magazine, vol.16 no. 10 (May 1997), p. 100-115.

J.M. Framiñan Torres: **Java**. Anaya Multimedia (col. Al Día en una Hora), Madrid, 1997. 128p.

Anexo 1

Anexo 1. Gestión del proyecto

Planificación del proyecto

A continuación se incluyen una tabla y un diagrama de Gantt en el que puede verse una planificación de los trabajos que se realizaron para el desarrollo del proyecto.

Nombre de tarea	Duración
-----------------	----------

Anexo 1: Anexo 1. Gestión del proyecto

PLANTEAMIENTO DE LA SOLUCIÓN	30 días
Recogida de requisitos en MERA	5 días
Definición arquitectura global del producto	10 días
Diseño de los módulos del producto	15 días
INVESTIGACIÓN	40 días
Elección de la tecnología para el desarrollo	10 días
Investigación sobre técnicas de análisis de información	15 días
Planteamiento de las creatividades	15 días
DESARROLLO DE LA SOLUCIÓN	90 días
Diseño de arquitectura del sistema	10 días
Desarrollo del componente de conexión a la BBDD	15 días
Desarrollo del software de negocio (módulos Contactos, Campañas, Newsletter...)	20 días
Desarrollo del software de administración (módulos Administración, Alertas,...)	10 días
Diseño de la solución de análisis (elección de las variables, programa matlab para los pesos, datos para entrenamiento...)	15 días
Desarrollo del software de análisis	10 días
Diseño y Desarrollo de la interfaz de acceso de usuario	20 días
DESARROLLO DE LA MEMORIA DEL PROYECTO	30 días
Desarrollo de documentación de usuario	5 días
Desarrollo de documentación del desarrollador	5 días
Desarrollo de Memoria del Proyecto	20 días

Tabla. 13: EDT (Estructura de Desglose de Tareas) del Proyecto

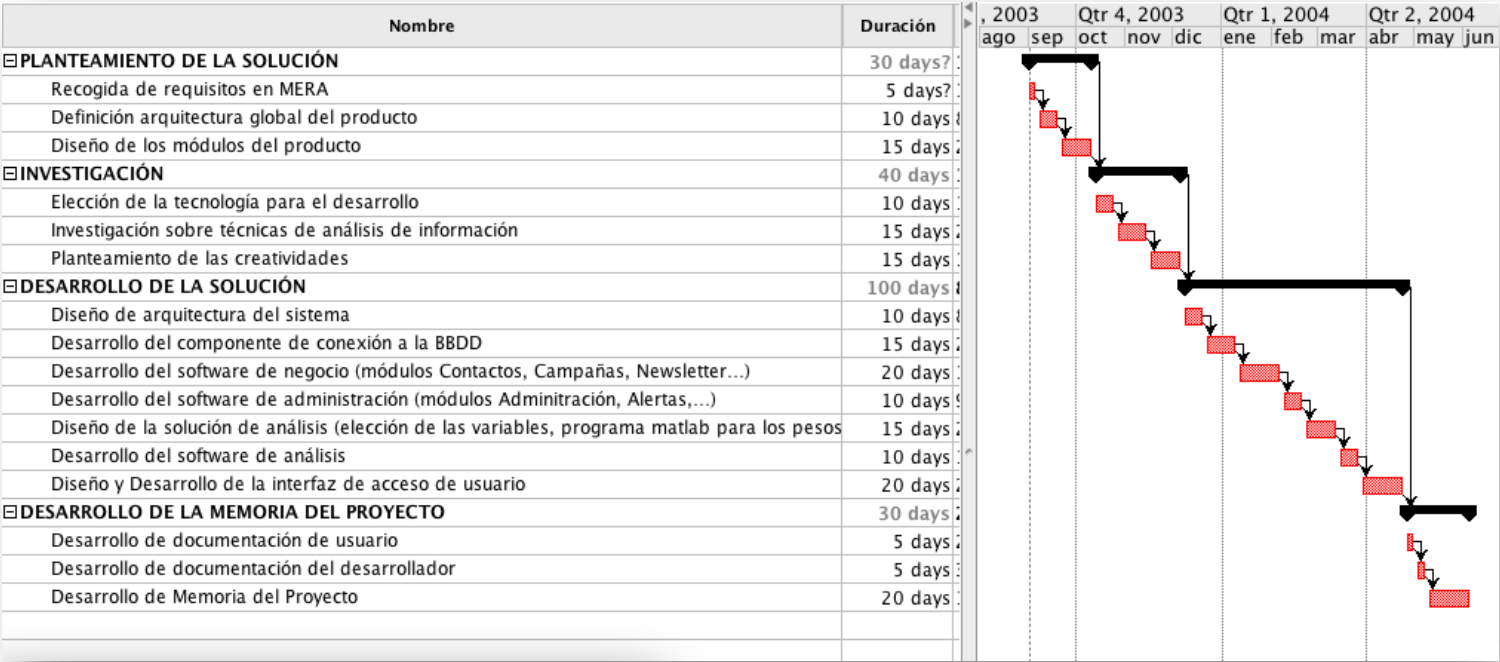


Ilustración 21: Diagrama Gantt Del Proyecto

Presupuesto

1) Partida servicios profesionales

Tabla 1: Partida de costes por perfiles profesionales

Perfil profesional	Núm. RR	€/Jornada	Num. Jorn.	Total
Tutor de proyecto	1	450 €	20	9.000 €
Desarrollador	1	285 €	190	54.150 €
			Subtotal	63.150 €

2) Sumatorio total de partidas de coste

Tabla 2: Sumatorio total de costes del proyecto

Concepto	Unidades	Importe
Partida coste mensual del entorno de desarrollo	9	9.000 €
Partida servicios profesionales	1	63.150 €
	Subtotal	72.150 €

(1) El presupuesto presentado no incluye el 21% de IVA.

(2) El presupuesto no incluye las partidas relacionadas con la creatividad.

Anexo 2

Anexo 2. Diseño Comercial del Sistema

Publicidad rmN 4, el CRM adaptado a sus necesidades

rmN 4, es la respuesta tecnológica a su necesidad de gestionar las interacciones con sus clientes a través de todos los canales (teléfono, carta, fax, sms, e-mail, web, etc.) y ofrecerles un servicio de valor añadido.

La implantación de rmN 4 le permite optimizar recursos, mejorar la comunicación interna y adquirir un conocimiento global del mercado y de las pretensiones de sus clientes siendo, por tanto, una herramienta esencial para incrementar sus ventas y optimizar su oferta de servicios.

rmN 4 se adapta a sus necesidades. Tanto si su negocio se centra en la venta de productos como si su actividad principal es la oferta de servicios o las actividades centradas en marketing, rmN 4 es la respuesta. Reduzca el tiempo invertido en el proceso de venta e incremente el promedio de pedidos, optimice la productividad de sus profesionales reduciendo costes y tiempo de respuesta al cliente, aumente sus índices de impacto por campaña, genere nuevas oportunidades de negocio y optimice la captación y fidelización de clientes. Deje que rmN 4 le muestre lo que puede hacer por su organización:

Anexo 2: Anexo 2. Diseño Comercial del Sistema

Generar más oportunidades de negocio	Gracias a las poderosas herramientas enfocadas a la creación, seguimiento y envío de campañas de marketing, tales como la elaboración de campañas de e-mail marketing con mensajes personalizados centrados en las preferencias de sus clientes.
Generar informes	Generar informes semanales o mensuales de forma automática, conservando la información necesaria sobre el rendimiento de sus clientes, así como pronósticos sobre los mismos, que le permitirán obtener análisis comparativos e históricos, fundamentales para asegurar la compatibilidad de las actuaciones de su organización con su estrategia de venta.
Mejora la efectividad de su servicio	rm ^N 4 le proporciona una flexible asignación de roles que asegura que cada incidencia será automáticamente dirigida a la persona adecuada, lo que permite reducir significativamente el tiempo de respuesta y satisfacer eficazmente las necesidades del cliente.
Reducción de tiempos de resolución de reclamaciones	Poniendo a disposición de sus profesionales el acceso instantáneo al expediente del cliente y a la base de datos donde se almacenan problemas comunes. Además, rm ^N 4 le permite actualizar dicha base de datos añadiendo nuevas soluciones a las ya existentes.
Incremento de resultados	Gracias a los análisis sobre efectividad de campañas, newsletters, SMS, portales, ventas por canal, etc. que le permitirán identificar eficazmente oportunidades de negocio así como potenciales amenazas y llevar a cabo acciones coordinadas en los diferentes equipos y departamentos.

Tabla. 14: Matriz de Funcionalidades

Otras ventajas de rmN:

- Optimiza los canales de relación con sus clientes: marketing, call centers, ventas, Internet, servicio postventa, etc.
- Aumenta su ventaja competitiva
- Aumenta sus ingresos y beneficios
- Proporciona un profundo análisis sobre ventas, productos y clientes
- Potencia la labor del equipo comercial
- Identifica el target de consumidores por producto
- Gestiona las oportunidades de negocio
- Agiliza los procesos de servicio (postventa, e-commerce, acciones sobre canal, etc.)

rm^N 4 es una solución flexible y robusta, verticalizada para trece sectores empresariales y avalada por la experiencia de Planet Media en la creación de soluciones inteligentes para empresas con visión de futuro.

Anexo 3

Anexo 3. Extractos de código fuente

Detalle de programación Módulo Análisis

En este código se ve parte del módulo de análisis. Concretamente éste método calcula, en base a los envíos de una campaña o newstetter, la cantidad de aperturas de la misma y los accesos a los vínculos, si la campaña se puede considerar un éxito o no en base a la función de pesos obtenida para Campañas.

```
public int getActivValue(int envios, int visitas, int clicks)
{
    double aux;
    int result;
    double maxhit= envios*5;
    double maxvisitas=envios*2;
    aux = (((visitas/maxvisitas)*0.4) + ((clicks/maxhit)*0.6));
    aux=aux*100;
    Double d = new Double (aux);
    result = d.intValue();

    if (result > 100)
        result = 100;

    return result;
}
```

```

public int getSessImp(int hits, int visitas,String fecha1, String fecha2){

    double aux;
    int result;
    double maxhit= 5;
    double maxvisitas=1;

    hits = hits/diffDate(fecha1,fecha2);
    visitas = visitas/diffDate(fecha1,fecha2);

    aux = (((visitas/maxvisitas)*0.4) + ((hits/maxhit)*0.6));

    aux=aux*100;
    Double d = new Double (aux);
    result = d.intValue();

    if (result > 100)
        result = 100;

    return result;

}

estrella = getActivValue((camps+news),visitas,hits);

estrellaS = getSessImp(hit,vis,fecha1,fecha2);

double star = estrella*(0.5) + estrellaS*(0.5);

star= 0.5*( (visitas/envios*2)*0.4 + (clicks/envios*5)*0.6 ) + ( (visitas/tiempo)*0.4 + ((clicks/tiempo)/5)

public int getExitoValue(int envios, int visitas, int clicks)
{
    double aux;
    int result;
    double maxhit= envios*5;
    double maxvisitas=envios*2;
    aux = (((visitas/maxvisitas)*0.4) + ((clicks/maxhit)*0.6));
    aux=aux*100;
    Double d = new Double (aux);
    result = d.intValue();

    if (result > 100)
        result = 100;

    return result;

}

public int getExitoValueExtranet(int visitas)
{

```

Anexo 3: Anexo 3. Extractos de código fuente

```
        visitas = (visitas * 100)/25;
        if (visitas > 100)
            visitas = 100;
        return visitas;
    }

    public int getActivValue(int envios, int visitas, int clicks)
    {
        double aux;
        int result;
        double maxhit= envios*5;
        double maxvisitas=envios*2;
        aux = (((visitas/maxvisitas)*0.4) + ((clicks/maxhit)*0.6));
        aux=aux*100;
        Double d = new Double (aux);
        result = d.intValue();

        if (result > 100)
            result = 100;

        return result;
    }

    public int getSessImp(int hits, int visitas,String fecha1, String fecha2){

        double aux;
        int result;
        double maxhit= 5;
        double maxvisitas=1;

        hits = hits/diffDate(fecha1,fecha2);
        visitas = visitas/diffDate(fecha1,fecha2);

        aux = (((visitas/maxvisitas)*0.4) + ((hits/maxhit)*0.6));

        aux=aux*100;
        Double d = new Double (aux);
        result = d.intValue();

        if (result > 100)
            result = 100;

        return result;
    }
```

Clases Implementadas

```

| +---planetmedia
| | +---rmn
| | | | FileHandler.java
| | | | FileHandlerRmn.java
| | | | Help.java
| | | | UserAdminRmn.java
| | | | UserManager.java
| | | |
| | | +---errors
| | | | rmNLogWriter.java
| | | |
| | | \---útil
| | | | RmnDbProp.java
| | | | RmnProp.java
| | | | RmnProperties.java
| | | |
| | | \---útil
| | | | ReadEnv.java
| | | | UserMail.java
| | | | databaseConfiguration.xml
| | | |
| | | +---BDControl
| | | | BDConfiguration.java
| | | | BDManager.java
| | | | BDPager.java
| | | | columnElement.java
| | | |
| | | +---error
| | | | Debug.java
| | | | LogWriter.java
| | | | PLErrorHandler.java
| | | | PSErrHandler.java
| | | |
| | | +---mail
| | | | | UserMail.java
| | | | |
| | | +---paginar
| | | | paginado.java
| | | |

```

Anexo 3: Anexo 3. Extractos de código fuente

```
| | \---session
| |     Session.java
```

Código Matlab

```

function [pesAdaline]=Adaline1(Xent,Yent,Xtst,Ytst, paso, maxVueltas);
% Implementación de la regla del Adaline

% Se establece un tope de vueltas para evitar un bucle infinito en casos de
% whos

we=rand(1,size(Xent,1)); %Vector de pesos inicial creado aleatoriamente
a = paso;                %Tasa de aprendizaje
n_err = 1;                %Inicialización del numero de errores sobre el conjunto de test dados unos pesos
lmsErr=0;
vueltas = 0;

% El proceso se repite mientras que haya errores de clasificación
% (y no se haya llegado al numero máximo de vueltas)

while n_err ~= 0 && vueltas < maxVueltas

    vueltas = vueltas +1;
    % Se aleatoriza el orden de las muestras en el algoritmo

    I = randperm(size(Xent,2));

    for i = 1:size(Xent,2)
        % El bucle debe interrumpirse cuando se haya alcanzado un vector
        % que produzca 0 errores de clasificación para el conjunto

        if(n_err == 0)
            break;
        end
        % Se limpia el contador de errores para cada muestra
        n_err = 0;
        % Dada la siguiente muestra, se consigue la salida "z" según el
        % vector de pesos actual
        xe = Xent(:,I(i));
        z = xe'*we';
        % Se calcula el error producido adaptando la salida al rango entre -1 y +1,
        % y en base a el se actualiza el vector de pesos
        err = (Yent(I(i)) - tanh(z));
        %err = (Yent(I(i)) - z);

        we = we + a*err*xe';

        % Se evalúa el nuevo vector sobre el conjunto total de datos

        for j=1:size(Xent,2)
            % Para cada muestra se calcula el error cometido
            xe = Xent(:,j);
            z = xe'*we';
            o = sign(z);
            err = Yent(j) - o;
            % En caso de existir error, se añade a la cuenta de errores
            % y se deja de evaluar el vector porque no es el optimo
            if(err ~= 0)
                n_err = n_err +1;
                break;
            end
        end
    end
end
end

```

Anexo 3: Anexo 3. Extractos de código fuente

```
% obtengo Tasas de error

n_err=0;

lmsErr=0;

for j=1:size(Xent,2)
% Para cada muestra se calcula el error cometido

    xe = Xent(:,j);
    z = xe'*we';
    o = sign(z);
    err = Yent(j) - o;
    lmsErr=lmsErr+(0.5*(Yent(j)-z)^2);

% En caso de existir error, se añade a la cuenta de errores
% y se deja de evaluar el vector porque no es el optimo

    if(err ~= 0)
        n_err = n_err +1;
    end
end

lmsErr=lmsErr/1000
n_err
lmsErr=0;
n_err=0;

for j=1:size(Xtst,2)
% Para cada muestra se calcula el error cometido

    xe = Xtst(:,j);
    z = xe'*we';
    o = sign(z);
    err = Ytst(j) - o;
    lmsErr=lmsErr+(0.5*(Ytst(j)-z)^2);

% En caso de existir error, se añade a la cuenta de errores
% y se deja de evaluar el vector porque no es el optimo

    if(err ~= 0)
        n_err = n_err +1;
    end
end

lmsErr=lmsErr/1000

n_err
we

m=we(2)/we(1)*(-1);
o=we(3)/we(1)*(-1);

%plot(sqrt(m*(-3:0.1:3)+o),-3:0.1:3,'g')
%plot((-1)*sqrt(m*(-3:0.1:3)+o),-3:0.1:3,'y')

pesAdaline=we;
```